

TUGAS AKHIR - 141599

**PERANCANGAN KONTROLER LQR - PID DALAM SISTEM
PENELUSURAN JALUR UNTUK *UNMANNED AERIAL
VEHICLES***

M. Dhyaksa Khairana
07111440000166

Dosen Pembimbing
Ir. Rusdhianto Effendie A.K, M.T.
Nurlita Gamayanti, ST., M.T.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018

Halaman ini sengaja dikosongkan



TUGAS AKHIR - 141599

**PERANCANGAN KONTROLER LQR - PID DALAM SISTEM
PENELUSURAN JALUR UNTUK *UNMANNED AERIAL
VEHICLES***

M. Dhyaksa Khairana
07111440000166

Dosen Pembimbing
Ir. Rusdhianto Effendie A.K, M.T.
Nurlita Gamayanti, ST., M.T.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018

Halaman ini sengaja dikosongkan



ITS

Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - 141599

***LQR - PID CONTROLLER DESIGN IN PATH FOLLOWING
SYSTEM FOR UNMANNED AERIAL VEHICLES***

M. Dhyaksa Khairana
07111440000166

Supervisor

Ir. Rusdhianto Effendie A.K, M.T.
Nurlita Gamayanti, ST., M.T.

***ELECTRICAL ENGINEERING DEPARTEMENT
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2018***

Halaman ini sengaja dikosongkan

**PERANCANGAN KONTROLER LQR-PID DALAM SISTEM
PENELUSURAN JALUR UNTUK UNMANNED AERIAL
VEHICLES**

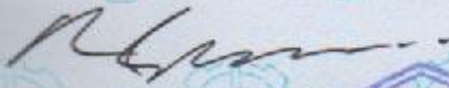
TUGAS AKHIR

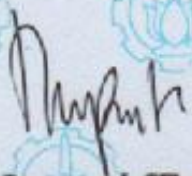
**Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I,

Dosen Pembimbing II,


Ir. Rusdhianto Effendi AK., M.T.
NIP. 19570424 198502 1001


Nurlita Gamavanti ST., M.T.
NIP. 19781201 200212 2002



PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Perancangan Kontroler LQR-PID dalam Sistem Penelusuran Jalur untuk *Unmanned Aerial Vehicles***” adalah merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2018

M. Dhyaksa Khairana
NRP 07111440000166

Halaman ini sengaja dikosongkan

PERANCANGAN KONTROLER LQR-PID DALAM SISTEM PENELUSURAN JALUR UNTUK UNMANNED AERIAL VEHICLES

Nama Mahasiswa : M. Dhyaksa Khairana
NRP : 07111440000166
Pembimbing : 1. Ir. Rusdhianto Effendie AK,MT.
2. Nurlita Gamayanti, ST., MT.

ABSTRAK

Sebuah pesawat tanpa awak (*Unmanned Aerial Vehicles*) perlu dikendalikan agar mampu terbang tanpa perlu bantuan seorang pilot. Salah satu caranya ialah dengan memberikan titik – titik tertentu pada koordinat agar pesawat melewati titik – titik tersebut dengan mengendalikan sudut moncong pesawat terhadap titik – titik tersebut. Pengendalian ini menjadi lebih rumit dikarenakan persamaan dinamika pesawat bersifat *non-linear* dan *coupled*. Penggabungan pengendali LQR dan PID merupakan salah satu metode untuk mengendalikan sudut agar pesawat dapat melewati referensi yang diberikan. Cara menggabungkannya ialah dengan menggunakan metode *LQR tuning*, yaitu di mana output dari LQR menghasilkan gain proporsional, integral, dan derivatif yang sesuai dengan kondisi sistem. Namun, sistem harus memiliki karakteristik *linear*. Sehingga diperlukan linearisasi agar pesawat dapat dilakukan *tuning* oleh LQR. Di dalam tugas akhir ini, dirancang sebuah sistem *path following* dengan *non-linear state feedback decoupler* dan kontroler *LQR tuning – PID*. Tipe referensi yang diberikan adalah titik pada koordinat (x, y). Hasil simulasi menunjukkan bahwa hasil *time constant* yang diinginkan masih sesuai dengan toleransi 0,1s yang diberikan untuk pengujian referensi pengendali. Dan untuk hasil simulasi *path following* memiliki *Root-Mean-Square-Error* (RMSE) 31,1735

Kata Kunci : *Unmanned Aerial Vehicles, Non-linear State Feedback Decoupler, LQR tuning – PID, Path Following*

Halaman ini sengaja dikosongkan

LQR-PID CONTROLLER DESIGN IN PATH FOLLOWING SYSTEM FOR UNMANNED AERIAL VEHICLES

By : M. Dhyaksa Khairana
Student Identity Number : 07111440000166
Supervisor(s) : 1. Ir. Rusdhianto Effendie AK,MT.
2. Nurlita Gamayanti, ST., MT.

ABSTRACT

Unmanned Aerial Vehicles (UAV) needs to be controlled so it can fly without pilot. One of the means is to add a few points in a coordinate so the plane will go through those points by controlling the heading angle of the plane. This get more complicated as the dynamic equation of the plane is non-linear and coupled. The combination of LQR and PID controller is one of the methods to control the heading angle of the plane so it can go to the specified reference. The way to combine both controllers is by using LQR tuning method, which is the output of LQR produce proportional, integral, and derivative gain that match the system. But to do that, the system has to be linear. So, the linearization method is needed so the plane could be tuned by LQR. In this book, path following system is designed with non-linear state feedback decoupler and LQR tuning – PID controller. The type of reference given is the coordinate points (x, y). The simulation results in all the controller is working good for constant input with tolerance of 0.1s. And for the path following system, the Root-Mean-Square-Error (RMSE) results in 31.1735.

Keywords : *Unmanned Aerial Vehicles, Non-linear State Feedback Decoupler, LQR tuning – PID, Path Following*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat dan karunianya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“PERANCANGAN KONTROLER LQR-PID DALAM SISTEM PENELUSURAN JALUR UNTUK *UNMANNED AERIAL VEHICLES*.”** Adapun tujuan dari penyusunan tugas akhir ini adalah sebagai salah satu persyaratan untuk menyelesaikan studi tahap sarjana pada bidang studi Teknik Sistem Pengaturan, Departemen Teknik Elektro, Fakultas Teknologi Elektro. Institut Teknologi Sepuluh Nopember.

Pada kesempatan ini, penulis mengucapkan banyak terima kasih pada pihak – pihak yang telah banyak berjasa terutama yang telah membantu menyelesaikan tugas akhir ini, antara lain:

1. Segenap keluarga tercinta yang berada di rumah. Ayah, Ibu, dan adik yang selalu tidak kenal lelah memberi doa, semangat, dan dukungan yang tiada henti untuk keberhasilan penulis.
2. Bapak Ir. Rusdhianto Effendie AK, MT dan Ibu Nurlita Gamayanti, ST., MT. Selaku dosen pembimbing yang telah memberikan saran dan bantuan sehingga tugas akhir ini dapat terselesaikan dengan baik.
3. Mas Khoiron Benraxid dan Muh. Alif Farabi yang telah memberikan banyak masukan dan saran selama penyelesaian tugas akhir ini.
4. Seluruh rekan dan teman – teman bidang studi Teknik Sistem Pengaturan angkatan 2014 lainnya yang telah menemani penulis selama masa perkuliahan dan penyelesaian tugas akhir di perguruan tinggi ini.
5. Seluruh rekan Laboratorium Sistem dan Sibernetika AJ204 dan B405 yang telah memberikan bantuan dan dukungan selama penulis menyelesaikan tugas akhir ini.
6. Seluruh keluargabesar Departemen Teknik Elektro, para dosen, dan karyawan atas dukungan dan kerjasamanya selama masa perkuliahan dan penyelesaian tugas akhir ini

Demikian yang dapat penulis sampaikan. Penulis menyadari masih banyaknya kekurangan dalam laporan tugas akhir ini. Oleh karena itu, penulis sangat terbuka dengan kritik dan saran dari pembaca laporan tugas

akhir ini. Terakhir, semoga laporan tugas akhir ini dapat bermanfaat untuk banyak pihak.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR	v
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
1.5 Metodologi.....	3
1.6 Sistematika Penulisan.....	4
1.7 Relevansi	5
BAB 2 TEORI PENUNJANG DAN DASAR TEORI.....	7
2.1 <i>Unmanned Aerial Vehicles</i> tipe <i>Fixed Wing</i>	7
2.2 Model Matematika UAV	8
2.2.1 Dinamika Longitudinal.....	8
2.2.2 Dinamika Lateral.....	9
2.2.3 Momen dan Gaya	12
2.2.4 Kumpulan Persamaan <i>Non-linear</i>	15
2.3 Metode Kontrol UAV.....	16
2.3.1 <i>Non-Linear State Feedback Decoupler</i>	17
2.3.2 Pengendali PID	18
2.3.4 <i>Linear Quadratic Regulator</i>	20
2.3.4 Pengendali <i>LQR tuning</i> - PID.....	21
2.3.5 Pengendali Ketinggian dengan <i>PID Cascade</i>	23
2.4 <i>Path Following</i>	24
2.4.1 <i>Path Generator</i>	24
2.4.2 <i>Guidance</i>	24
2.4.2.1 <i>Line of Sight</i>	25
2.4.2.2 <i>Circle of Acceptance</i>	26
BAB 3 PERANCANGAN SISTEM.....	27
3.1 Gambaran Umum Sistem	27
3.2 Perancangan Model UAV <i>Fixed Wing</i>	27

3.3	Perancangan <i>Non-Linear State Feedback Decoupler</i>	31
3.4	Perancangan Kontroler <i>LQR tuning</i> - PID	34
3.5	Perancangan <i>Altitude Control</i>	37
3.6	Perancangan <i>Waypoint Planner</i>	37
BAB 4	HASIL DAN ANALISIS.....	39
4.1	Gambaran Umum Pengujian Simulasi	39
4.2	Pengujian Model Dinamika UAV	40
4.3	Pengujian <i>Waypoint Planner</i>	43
4.4	Pengujian Pengendali <i>LQR tuning</i> - PID	45
4.5	Pengujian <i>Altitude Control</i>	48
4.4	Pengujian Sistem Path Following	50
BAB 5	KESIMPULAN DAN SARAN	55
5.1	Kesimpulan	55
5.2	Saran	56
DATAR PUSTAKA.....		57
LAMPIRAN.....		59
A1.	Diagram Blok Sistem.....	59
A2.	Program Dinamika UAV	59
A3.	Program <i>Non-linear State Feedback Decoupler</i>	64
A4.	Program <i>LQR tuning</i> – PID	67
A4.1	<i>Roll</i>	67
A4.2	<i>Pitch</i>	68
A4.3	<i>Yaw</i>	68
A5.	Program <i>Waypoint Planner</i>	68
A5.1	<i>Path Generator</i>	69
A5.2	<i>Guidance</i>	69
A6.	Program <i>Cross Track Error</i>	70
RIWAYAT HIDUP PENULIS.....		73

DAFTAR GAMBAR

Gambar 2.1 <i>Roll, pitch, dan yaw</i> pada pesawat.....	7
Gambar 2.2 Efek distribusi tekanan pada UAV	12
Gambar 2.3 Diagram blok <i>Non-Linear State Feedback Decoupler</i>	18
Gambar 2.4 Arsitektur pengendali PID	19
Gambar 2.5 Diagram blok <i>LQR tuning – PID</i>	21
Gambar 2.6 Teori Konfigurasi <i>Cascade</i>	23
Gambar 2.7 <i>Line of Sight</i>	24
Gambar 2.8 Circle of Acceptance	26
Gambar 3.1 Penambahan saturasi pada kontroler PID untuk sudut yaw	37
Gambar 3.2 Altitude Control	37
Gambar 4.1 Respon <i>roll</i> tanpa adanya pengendali.....	40
Gambar 4.2 Respon <i>pitch</i> tanpa adanya pengendali	41
Gambar 4.3 Respon <i>yaw</i> tanpa adanya pengendali	41
Gambar 4.4 Posisi x terhadap bumi tanpa adanya pengendali	42
Gambar 4.5 Posisi y terhadap bumi tanpa adanya pengendali	42
Gambar 4.6 Posisi ketinggian tanpa adanya pengendali	43
Gambar 4.7 Titik – titik <i>waypoint</i>	44
Gambar 4.8 <i>Path</i> untuk UAV	45
Gambar 4.9 Respon <i>roll</i> pengendali <i>LQR tuning – PID</i> referensi 0,2 .	46
Gambar 4.10 Respon <i>pitch</i> pengendali <i>LQR tuning – PID</i> referensi 0,2	47
Gambar 4.11 Respon <i>yaw</i> pengendali <i>LQR tuning – PID</i> referensi 0,2	47
Gambar 4.12 Respon pengendali ketinggian dengan referensi +20m.	46
Gambar 4.13 Respon <i>yaw</i> setelah penambahan saturasi dengan referensi 0,2.....	50
Gambar 4.14 Sistem <i>path following</i>	51
Gambar 4.15 Perbandingan jalur UAV dengan <i>path generator</i>	52
Gambar 4.16 Grafik <i>cross track error</i>	52

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Karakteristik komponen dalam PID	20
Tabel 3.1 Parameter UAV	29
Tabel 3.2 Estimasi parameter UAV	30
Tabel 4.1 Titik <i>waypoint</i>	44
Tabel 4.2 Nilai gain PID dari <i>tuning</i> LQR	46
Tabel 4.3 Data hasil pengujian untuk spesifikasi kontroler	48
Tabel 4.4 Hasil pengujian ketinggian	49
Tabel 4.5 Perubahan τ pada respon <i>yaw</i>	51

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Bab ini menjelaskan latar belakang dari penulisan tugas akhir ini, perumusan masalah yang ditentukan, batasan – batasan masalah yang ada, tujuan penelitian tugas akhir, metodologi pelaksanaan tugas akhir, sistematika penulisan laporan tugas akhir, dan relevansi.

1.1 Latar Belakang

Saat ini, perkembangan *vehicles automation* sedang mengalami perkembangan yang cukup pesat. Dari *autonomous car* hingga *unmanned aerial vehicles*. Di dalam buku ini, penulis memfokuskan diri di dalam sistem kontrol *unmanned aerial vehicles* dengan tipe *fixed wing*. Tipe *fixed wing* ialah di mana sebuah *aircraft* memiliki komponen steering dasar berupa *aileron*, *elevator*, dan *rudder*. Dalam proses agar sebuah *aircraft* dapat dijalankan secara otomatis, terdapat tahap-tahap yang harus dilakukan dan dipertimbangkan.

Di dalam buku ini, permasalahan yang diteliti ialah sistem penulurusan jalur (*path following*) dari sebuah *unmanned aerial vehicles*. Yaitu di mana referensi masukan ke sistem berupa jalur di koordinat cartesian. Variabel yang dikontrol ialah sudut *heading* dari pesawat agar pesawat dapat mengarah ke titik referensi agar *error* selalu mendekati nol. Hal ini dapat dilakukan dengan memanfaatkan gaya – gaya aerodinamika, dikarenakan perubahan sudut pada komponen *aileron*, *elevator*, dan *rudder*.

Metode kontrol yang dilakukan ialah menggabungkan kontrol optimal dengan kontrol PID. Kontroler PID ialah kontroler yang memiliki komponen proporsional, integral, dan derivatif. Di mana *gain proportional* akan mempersingkat *rise time* dan meningkatkan *overshoot*, *gain integral* akan mempersingkat *rise time* dan menghilangkan *steady state error*, dan *gain derivative* akan mengurangi *overshoot* dan osilasi dari *output*. Terdapat beberapa metode untuk menentukan *gain* proporsional, integral, dan derivatif. Salah satu diantaranya ialah metode Ziegler-Nichols. Namun, di dalam buku ini, metode *tuning gain* untuk komponen proporsional, integral, dan derivatif didapatkan dari metode kontrol optimal LQR (*Linear Quadratic Regulator*). Dikarenakan metode

tuning yang dilakukan ialah LQR-tuning PID, harus didapatkan bentuk *linear* dari model matematika UAV yang berbentuk *non-linear*.

1.2 Permasalahan

Dalam pergerakan pesawat, *waypoint planner* menjadi referensi *input* dari seluruh sistem. Seiring berjalannya waktu, pesawat akan berjalan dari satu titik ke titik lainnya. Oleh karena itu diperlukan sebuah kendali yang dapat mengontrol aircraft dengan strategi mengontrol sudut heading. Jenis kontroler yang dipilih ialah LQR-tuning PID. Dengan menggabungkan dua jenis kontroler, diharapkan agar mendapatkan keuntungan dari kedua kontroler ini. Namun, dikarenakan menggunakan metode LQR untuk mendapatkan *gain* PID, sistem yang dikendalikan harus dilinearkan terlebih dahulu.

1.3 Batasan Masalah

Beberapa batasan masalah dalam penelitian ini adalah sebagai berikut.

- Penelitian ini menggunakan tipe UAV *fixed wing*.
- Nilai percepatan gravitasi bumi dan densitas udara dianggap konstan.
- Pesawat memiliki 6 *degrees of freedom* (6-DOF).
- Dalam pergerakannya, kecepatan tidak termasuk variabel yang dikendalikan.
- Persamaan propulsi tidak termasuk dalam sistem yang mengakibatkan tidak ada perubahan kecepatan.
- Informasi yang ada bersifat deterministik
- Sistem disimulasikan dalam kondisi ideal

1.4 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah membuat suatu pengendali agar UAV dapat mengikuti jalur yang sudah diberikan oleh *waypoint planner* dengan kesalahan antara orientasi *aircraft* dengan orientasi referensi sekecil mungkin. Maka dari itu digunakan kontroler LQR-tuning PID untuk mengendalikan pesawat melewati *path*.

Sistem yang dikendalikan harus diubah ke bentuk *linear* terlebih dahulu dikarenakan untuk mendapatkan *gain* PID, digunakan metode

LQR. Pesawatnya sendiri memiliki karakteristik *non-linear*. Untuk melinearisasikan plant, digunakan metode *non-linear state feedback decoupler*. Dengan demikian, metode LQR dapat digunakan.

1.5 Metodologi

Tahapan penyelesaian tugas akhir ini adalah sebagai berikut.

- a) Studi Literatur
Melakukan studi literatur dalam mencari model matematika dari pesawat dan kontroler LQR-*tuning* PID. Sumber literatur berasal dari buku, jurnal ilmiah, serta catatan perkuliahan.
- b) Pemodelan Matematis UAV *Fixed Wing*
Membuat model matematis *non-linear* dari pesawat yang didapat dari sumber literatur. Lalu, model ini dijadikan kajian dalam tugas akhir ini.
- c) Linearisasi Persamaan Matematis Pesawat
Linearisasi dilakukan dengan metode *non-linear state feedback decoupler*. Hal ini dilakukan agar menghilangkan sifat-sifat *non-linear* dari pesawat.
- d) Perancangan Kontroler PID dengan LQR-*tuning*
Melakukan perancangan pengendali PID dengan metode *tuning* LQR (*Linear Quadratic Regulator*) untuk mengatur sudut *heading* dari pesawat. Pengujiannya dilakukan dengan pertama – tama memberikan *input* konstan. Lalu, akan dikembangkan dengan memiliki *input* sebuah *path*. Pengendalian dikatakan berhasil saat pesawat dapat mengikuti jalur yang diberikan.
- e) Perancangan *Waypoint Planner* serta *Altitude Control*
Waypoint planner digunakan sebagai pembuat *path* atau jalur yang harus dilalui pesawat. *Waypoint* diberikan dengan memberi titik – titik di posisi tertentu dalam koordinat cartesian. Lalu, diantara titik – titik tersebut akan ditarik sebuah garis dengan metode interpolasi agar membentuk sebuah jalur. Sedangkan untuk *altitude control* digunakan untuk mengendalikan ketinggian dari pesawat dengan memberikan referensi berupa ketinggian dalam meter dan mengkonfigurasi kontrolernya secara cascade dengan pengendali sudut *pitch*.

- f) Hasil Simulasi dan Analisis
Setelah semua tahap dari (a) ke (e) dilakukan, dilakukan simulasi pesawat mengikuti jalur yang diberikan. Simulasi dilakukan dengan perbedaan masukan massa konstan dan massa berubah terhadap waktu.
- g) Kesimpulan dan Saran
Penarikan kesimpulan dilakukan setelah melakukan analisis dari simulasi yang telah dijalankan sebelumnya. Selanjutnya pemberian saran berupa seperti apa penelitian yang dapat dilanjutkan dari tugas akhir ini.
- h) Penyusunan Buku Tugas Akhir
Penulisan buku laporan tugas akhir dilakukan sebagai bentuk laporan tertulis dari pelaksanaan tugas akhir ini.

1.6 Sistematika Penulisan

Laporan tugas akhir ini dibagi menjadi lima bab dengan uraian singkat masing – masing bab diberikan sebagai berikut.

BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang dari pelaksanaan tugas akhir, permasalahan yang ada, tujuan pelaksanaan tugas akhir, metodologi pelaksanaan, sistematika penulisan laporan tugas akhir, dan relevansi tugas akhir ini.

BAB II TINJAUAN PUSTAKA

Dalam bab ini dijelaskan teori – teori penunjang yang akan digunakan dalam penyelesaian tugas akhir ini meliputi transformasi koordinat, model matematika pesawat, *path following*, linearisasi dengan metode *decoupler*, kontroler PID, *Linear Quadratic Regulator*, dan kontroler *LQR-tuning PID*.

BAB III PERANCANGAN SISTEM

Bab ini berisi dengan penjelasan langkah – langkah pembuatan model matematika pesawat, perancangan linearisasi *decoupler*, perancangan kontroler *LQR tuning-PID*, dan perancangan *waypoint planner*.

BAB IV HASIL DAN ANALISIS

Di dalam bab ini terdapat pengujian dari simulasi pesawat ketika referensi berupa konstan, dan referensi berupa *path*.

BAB V PENUTUP

Bab ini berisi kesimpulan yang diambil dari hasil simulasi beserta analisis untuk simulasi dan saran untuk pelaksanaan penelitian selanjutnya dari tugas akhir ini.

1.7 Relevansi

Hasil dari pelaksanaan tugas akhir ini diharapkan dapat memberikan manfaat sebagai salah satu referensi yang dapat digunakan dalam penelitian-penelitian berikutnya mengenai penggabungan kontroler optimal dan kontroler PID dan penelitian mengenai sistem *Unmanned Aerial Vehicles*. Selain itu dapat membuka topik kajian baru dengan menggunakan spesifikasi sistem yang berbeda.

Halaman ini sengaja dikosongkan

BAB 2

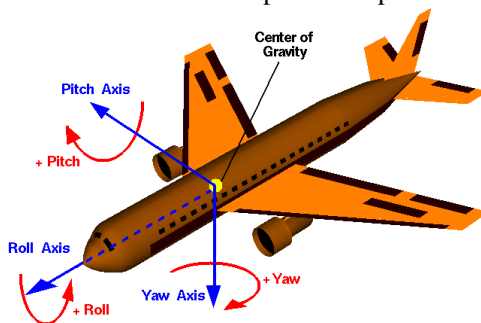
TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai dasar teori yang digunakan dalam pelaksanaan tugas akhir ini, yaitu mengenai model matematika dari UAV, *waypoint planner*, sistem linearisasi, serta metode kontrol yang dilakukan.

2.1 *Unmanned Aerial Vehicles tipe Fixed Wing*

Unmanned Aerial Vehicles atau yang biasa disingkat UAV adalah pesawat terbang tanpa awak. Tipe *fixed wing* merupakan salah satu bentuk fisik dari UAV. Bentuk *fixed wing* ialah di mana pesawat menggunakan gaya aerodinamis dari *body* untuk bermaneuver di udara. Komponen yang membuat pesawat dapat bermaneuver adalah sirip-sirip yang disebut *aileron*, *elevator*, dan *rudder*. *Aileron* digunakan untuk mengatur sudut *roll* dari pesawat, *elevator* digunakan untuk mengatur sudut *pitch*, dan *rudder* digunakan untuk mengatur sudut *yaw*.

Roll, *pitch*, dan *yaw* merupakan notasi putar yang digunakan untuk sudut dalam sumbu x , y , z . Sudut ini juga bisa disebut dengan sudut Euler. Sudut Euler adalah merupakan salah satu metode transformasi sistem koordinat yang umum digunakan dalam penerbangan maupun simulasi dari pesawat udara. Sudut Euler digunakan untuk menunjukkan hubungan sistem koordinat yang bergerak dengan sistem koordinat yang tetap. Ilustrasi dari sudut Euler dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Roll, pitch, dan yaw* pada pesawat[9]

2.2 Model Matematika UAV

Terdapat langkah – langkah yang harus dilakukan sebelum kita dapat mengaplikasikan sebuah kontroler ke dalam pesawat. Langkah terpenting ialah mengetahui karakteristik dan respon pesawat terhadap beberapa gaya yang berpengaruh terhadap body. Maka dari itu, kita harus mendapatkan model dinamika pesawat baik longitudinal maupun lateral. Serta transformasi koordinat body ke koordinat bumi. Untuk memodelkan pesawat, dapat diturunkan dari persamaan hukum Newton II.

2.2.1 Dinamika Longitudinal[1]

Dalam memodelkan dinamika longitudinal, digunakan hukum Newton II. Secara umum diturunkan sebagai berikut

$$m \frac{d\mathbf{V}_g}{dt_i} = \mathbf{f} \quad (2.4)$$

Di mana m adalah massa pesawat, \mathbf{V}_g adalah vector kecepatan kendaraan pada tiga sumbu $(u, v, w)^T$, $\frac{d}{dt_i}$ adalah derivatif terhadap *earth frame*, dan \mathbf{f} adalah total gaya luar yang bekerja pada kendaraan. Lalu semua akan dipindah ke *body frame* agar lebih mudah dianalisa. Persamaan (2.4) akan diubah ke *body frame* dengan menggunakan persamaan

$$\frac{d\mathbf{V}_g}{dt_i} = \frac{d\mathbf{V}_g}{dt_b} + \omega_{b/i} \times \mathbf{V}_g \quad (2.5)$$

Dengan mensubstitusikan Persamaan (2.4) dan Persamaan (2.5) maka diperoleh persamaan dinamika translasi pada *body frame*

$$m \left(\frac{d\mathbf{V}_g}{dt_b} + \omega_{b/i} \times \mathbf{V}_g \right) = \mathbf{f} \quad (2.6)$$

Persamaan (2.6) bisa diekspresikan sebagai berikut

$$\frac{d\mathbf{V}_g}{dt_b} = -\omega_{b/i} \times \mathbf{V}_g + \frac{1}{m} \mathbf{f} \quad (2.7)$$

di mana,

$$\frac{d\mathbf{V}_g}{dt_b} = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}, \omega_{b/i} = \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \mathbf{f} = \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}$$

Persamaan (2.7) dapat ditulis ulang menjadi model longitudinal pesawat

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \quad (2.8)$$

2.2.2 Dinamika Lateral[1]

Dalam memodelkan model dinamika lateral, digunakan hukum Newton II tentang *angular momentum*.

$$\frac{d\mathbf{h}}{dt_i} = \mathbf{m} \quad (2.9)$$

Di mana \mathbf{h} merupakan vektor momentum sudut dan \mathbf{m} adalah total semua momen yang bekerja dari luar, $(m, l, n)^T$. Mirip dengan penurunan dinamika longitudinal, dinamika lateral juga dirubah ke dalam *body frame*, menggunakan persamaan.

$$\frac{d\mathbf{h}}{dt_i} = \frac{d\mathbf{h}}{dt_b} + \omega_{b/i} \times \mathbf{h} \quad (2.10)$$

Lalu substitusikan Persamaan (2.9) ke Persamaan (2.10) agar diperoleh persamaan dinamika rotasi pada *body frame*.

$$\frac{d\mathbf{h}}{dt_b} + \omega_{b/i} \times \mathbf{h} = \mathbf{m} \quad (2.11)$$

Di mana untuk *rigid body*, momentum sudut merupakan perkalian antara matriks inersia dan kecepatan sudut.

$$\mathbf{h} = \mathbf{J}\omega_{b/i} \quad (2.12)$$

dengan nilai \mathbf{J} diberikan oleh

$$\begin{aligned} \mathbf{J} &= \begin{pmatrix} \int (y^2 + z^2) dm & -\int xy dm & -\int xz dm \\ -\int xy dm & \int (x^2 + z^2) dm & -\int yz dm \\ -\int xz dm & -\int yz dm & \int (x^2 + y^2) dm \end{pmatrix} \\ &\triangleq \begin{pmatrix} J_x & -J_{xy} & -J_{xz} \\ -J_{xy} & J_y & -J_{yz} \\ -J_{xz} & -J_{yz} & J_z \end{pmatrix} \end{aligned} \quad (2.13)$$

Hasil substitusi Persamaan (2.9) dan (2.10) dapat ditulis sebagai ekspresi dari percepatan sudut sebagai berikut

$$\mathbf{J} \frac{d\omega_{b/i}}{dt_b} + \omega_{b/i} \times \mathbf{h} = \mathbf{m} \Leftrightarrow \dot{\omega}_{b/i} = \mathbf{J}^{-1} [-\omega_{b/i} \times (\mathbf{J}\omega_{b/i}) + \mathbf{m}] \quad (2.14)$$

Dikarenakan pesawat simetris hanya pada bidang yang dibentang oleh sumbu-x dan sumbu-z, sehingga $J_{xy} = J_{yz} = 0$, mengakibatkan matriks inersia menjadi

$$\mathbf{J} = \begin{pmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{pmatrix} \quad (2.15)$$

Dengan asumsi ini didapatkan invers dari \mathbf{J} adalah

$$\mathbf{J}^{-1} = \frac{\begin{pmatrix} J_y J_z & 0 & J_y J_{xz} \\ 0 & J_x J_z - J_{xz}^2 & 0 \\ J_y J_{xz} & 0 & J_x J_y \end{pmatrix}}{J_x J_y J_z - J_{xz}^2 J_y} = \begin{pmatrix} \frac{J_z}{\Gamma} & 0 & \frac{J_{xz}}{\Gamma} \\ 0 & \frac{1}{J_y} & 0 \\ \frac{J_{xz}}{\Gamma} & 0 & \frac{J_x}{\Gamma} \end{pmatrix} \quad (2.16)$$

di mana $\Gamma \triangleq J_z J_x - J_{xz}^2$.

Sehingga model dapat ditulis menjadi

$$\begin{pmatrix} \ddot{p} \\ \ddot{q} \\ \ddot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_z}{\Gamma} & 0 & \frac{J_{xz}}{\Gamma} \\ 0 & \frac{1}{J_y} & 0 \\ \frac{J_{xz}}{\Gamma} & 0 & \frac{J_x}{\Gamma} \end{pmatrix} \begin{bmatrix} J_{xz} p q + (J_y - J_z) q r \\ J_{xz} (r^2 - p^2) + (J_z - J_x) p r \\ (J_x - J_y) p q - J_{xz} q r \end{bmatrix} + \begin{pmatrix} m \\ l \\ n \end{pmatrix} \quad (2.17)$$

di mana,

$$\Gamma \triangleq J_z J_x - J_{xz}^2$$

$$\Gamma_1 = \frac{J_{xz} (J_x - J_y + J_z)}{\Gamma}$$

$$\Gamma_2 = \frac{J_z (J_z - J_y) + J_{xz}^2}{\Gamma}$$

$$\Gamma_3 = \frac{J_z}{\Gamma}$$

$$\Gamma_4 = \frac{J_{xz}}{\Gamma}$$

$$\Gamma_5 = \frac{J_z - J_x}{\Gamma}$$

$$\Gamma_6 = \frac{J_{xz}}{J_y}$$

$$\Gamma_7 = \frac{J_x (J_x - J_y) + J_{xz}^2}{\Gamma}$$

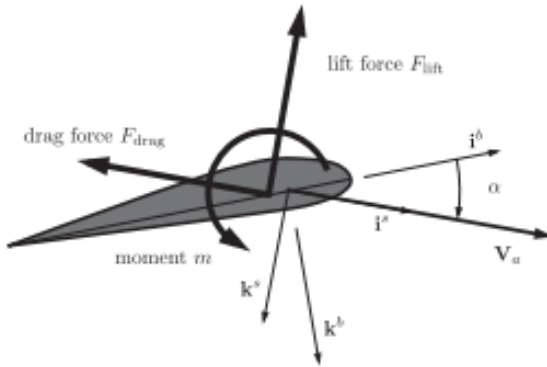
$$\Gamma_8 = \frac{J_x}{\Gamma}$$

Yang mengakibatkan persamaan dinamika lateral pesawat sebagai berikut

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \Gamma_1 p q - \Gamma_2 q r + \Gamma_3 l + \Gamma_4 n \\ \Gamma_5 p r - \Gamma_6 (p^2 - r^2) + \frac{1}{I_y} m \\ \Gamma_7 p q - \Gamma_1 q r + \Gamma_4 l + \Gamma_8 n \end{pmatrix} \quad (2.18)$$

2.2.3 Momen dan Gaya[1]

Ketika pesawat terbang, arah heading pesawat tidak selalu sama dengan arah gerak pesawat. Perbedaan ini dinotasikan sebagai *angle of attack* dan *slip angle*. Gaya – gaya yang dianalisa berada pada frame gerak pesawat yang disebut dengan *air frame*, yang kemudian diproyeksikan ke *body frame* menggunakan trigonometri.



Gambar 2.2 Efek distribusi tekanan pada UAV[1]

Analisis dilakukan pertama-tama pada sumbu-x dan sumbu-z dari body frame dikarenakan berhubungan langsung pada gerak longitudinal dari pesawat. Gaya – gaya ini merupakan hasil dari gaya *lift* dan gaya *drag* yang berada pada *air frame* seperti pada Gambar 2.2 Gaya *lift* dan gaya *drag* merupakan efek dari distribusi tekanan disekitar sayap pesawat dan diberikan oleh persamaan berikut

$$F_{lift} = \frac{1}{2} \rho V_a^2 S C_L \quad (2.19)$$

$$F_{drag} = \frac{1}{2} \rho V_a^2 S C_D \quad (2.20)$$

Di mana C_L dan C_D adalah koefisien aerodinamis, S adalah luas penampang sayap pesawat. C_L dan C_D dipengaruhi oleh kecepatan rotasi, q , defleksi *elevator*, δ_e , dan *angle of attack*, hubungan tersebut menggunakan pendekatan ekspansi taylor orde pertama dan mempertimbangkan ke-non-dimensional-an dari koefisien didapatkan formula koefisien aerodinamis sebagai berikut

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_q} \frac{c}{2V_a} q + C_{L_{\delta_e}} \delta_e \quad (2.21)$$

$$C_D = C_{D_0} + C_{D_\alpha} \alpha + C_{D_q} \frac{c}{2V_a} q + C_{D_{\delta_e}} \delta_e \quad (2.22)$$

Karena gaya dan gaya drag berada pada *air frame*, untuk mendapatkan nilainya pada *body frame* perlu dilakukan operasi trigonometri sebagai berikut

$$\begin{pmatrix} f_x \\ f_z \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} -F_{drag} \\ -F_{lift} \end{pmatrix} \quad (2.23)$$

Dengan operasi Persamaan (2.23), didapatkan gaya – gaya pada sumbu-x dan sumbu-z di *body frame*. Pada sumbu-x *body frame*, selain gaya yang diberikan oleh Persamaan (2.23), terdapat juga gaya yang diberikan oleh propulsi dengan persamaan sebagai berikut

$$f_{p_x} = \frac{1}{2} \rho S_{prop} C_{prop} ((k_{motor} \delta_t)^2 - V_a^2) \quad (2.24)$$

Sedangkan gaya pada sumbu-y *body frame* diberikan oleh persamaan

$$f_y = \frac{1}{2} \rho V_a^2 S C_Y \quad (2.25)$$

Di mana C_Y merupakan koefisien aerodinamis yang besarnya dipengaruhi oleh *slip angle*, β , kecepatan *roll*, p , kecepatan *yaw*, r , serta defleksi *aileron*, δ_a , dan defleksi *rudder*, δ_r . Menggunakan pendekatan ekspansi taylor orde pertama didapatkan hubungan antara variabel – variabel tersebut sebagai berikut

$$C_Y = C_{Y_0} + C_{Y_\beta} \beta + C_{Y_p} \frac{b}{2V_a} p + C_{Y_r} \frac{b}{2V_a} r + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \quad (2.26)$$

Di mana b adalah bentang sayap pesawat. Sedangkan, untuk dinamika lateral, notasi momen menggunakan $(l, m, n)^T$, secara berurutan adalah momen *rolling*, *pitching*, dan *yawing*. Besaran dari momen diberikan oleh persamaan

$$l = \frac{1}{2} \rho V_a^2 S b C_l(\beta, p, r, \delta_a, \delta_r) \quad (2.27)$$

$$m = \frac{1}{2} \rho V_a^2 S c C_m(\alpha, q, \delta_e) \quad (2.28)$$

$$n = \frac{1}{2} \rho V_a^2 S b C_n(\beta, p, r, \delta_a, \delta_r) \quad (2.29)$$

Di mana c adalah *mean cord* dari sayap pesawat. C_l , C_m , C_n , adalah koefisien aerodinamis yang dipengaruhi oleh variabel – variabel yang terdapat di dalam kurung, menggunakan pendekatan ekspansi taylor didapatkan persamaan berikut

$$C_l = C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{bp}{2V_a} + C_{l_r}\frac{br}{2V_a} + C_{l_{\delta a}}\delta_a + C_{l_{\delta r}}\delta_r \quad (2.30)$$

$$C_m = C_{m_0} + C_{m_\alpha} + C_{m_q}\frac{cq}{2V_a} + C_{m_{\delta e}}\delta_e \quad (2.31)$$

$$C_n = C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{bp}{2V_a} + C_{n_r}\frac{br}{2V_a} + C_{n_{\delta a}}\delta_a + C_{n_{\delta r}}\delta_r \quad (2.32)$$

Lalu disubstitusikan Persamaan (2.27), (2.28), dan (2.29) ke Persamaan (2.18).

2.2.4 Kumpulan Persamaan *Non-linear*[1]

Setelah didapatkan semua model dinamika baik pada longitudinal dan lateral, serta momen dan gaya, bisa kita kumpulkan semuanya menjadi persamaan percepatan seperti berikut

Longitudinal

$$\dot{u} = rv - qw - g \sin \theta + \frac{\rho V_a^2 S}{2m} \left[C_X(\alpha) + C_{X_q}(\alpha) \frac{cq}{2V_a} + C_{X_{\delta e}}(\alpha) \delta_e \right] + \frac{\rho S_{prop} C_{prop}}{2m} [(k_{motor} \delta_t)^2 - V_a^2] \quad (2.33a)$$

$$\dot{v} = pw - ru - g \cos \theta \sin \phi + \frac{\rho V_a^2 S}{2m} \left[C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{bp}{2V_a} + C_{Y_r}\frac{br}{2V_a} + C_{Y_{\delta a}}\delta_a + C_{Y_{\delta r}}\delta_r \right] \quad (2.33b)$$

$$\dot{w} = qu - pv - g \cos \theta \cos \phi + \frac{\rho V_a^2 S}{2m} \left[C_Z(\alpha) + C_{Z_q}(\alpha) \frac{cq}{2V_a} + C_{Z_{\delta e}}(\alpha) \delta_e \right] \quad (2.33c)$$

di mana,

$$C_X(\alpha) \triangleq -C_D(\alpha) \cos \alpha + C_L(\alpha) \sin \alpha$$

$$C_{X_q}(\alpha) \triangleq -C_{D_q} \cos \alpha + C_{L_q} \sin \alpha$$

$$C_{X_{\delta e}}(\alpha) \triangleq -C_{D_{\delta e}} \cos \alpha + C_{L_{\delta e}} \sin \alpha$$

$$C_Z(\alpha) \triangleq -C_D(\alpha) \sin \alpha - C_L(\alpha) \cos \alpha$$

$$C_{Z_q}(\alpha) = -C_{D_q} \sin \alpha - C_{L_q} \cos \alpha$$

$$C_D(\alpha) = C_{D_0} + C_{D_\alpha} \alpha \text{ dan } C_L(\alpha) = C_{L_0} + C_{L_\alpha} \alpha$$

Lateral

$$\dot{p} = \Gamma_1 pq - \Gamma_2 qr + \frac{\rho V_a^2 S b}{2} \left[C_{p_0} + C_{p_\beta}\beta + C_{p_p}\frac{bp}{2V_a} + C_{p_r}\frac{br}{2V_a} + C_{p_{\delta a}}\delta_a + C_{p_{\delta r}}\delta_r \right] \quad (2.34a)$$

$$\dot{q} = \Gamma_5 pq - \Gamma_6 (p^2 - r^2) + \frac{\rho V_a^2 S c}{2 l_y} \left[C_{m_0} + C_{m_a} + C_{m_q} \frac{cq}{2V_a} + C_{m_{\delta e}} \delta_e \right] \quad (2.34b)$$

$$\dot{r} = \Gamma_7 pq - \Gamma_8 qr + \frac{\rho V_a^2 S b}{2} \left[C_{r_0} + C_{r_\beta} \beta + C_{r_p} \frac{bp}{2V_a} + C_{r_r} \frac{br}{2V_a} + C_{r_{\delta a}} \delta_a + C_{r_{\delta r}} \delta_r \right] \quad (2.34c)$$

di mana,

$$\begin{aligned} C_{p_0} &= \Gamma_3 C_{l_0} + \Gamma_4 C_{n_0} \\ C_{p_\beta} &= \Gamma_3 C_{l_\beta} + \Gamma_4 C_{n_\beta} \\ C_{p_p} &= \Gamma_3 C_{l_p} + \Gamma_4 C_{n_p} \\ C_{p_r} &= \Gamma_3 C_{l_r} + \Gamma_4 C_{n_r} \\ C_{p_{\delta a}} &= \Gamma_3 C_{l_{\delta a}} + \Gamma_4 C_{n_{\delta a}} \\ C_{p_{\delta r}} &= \Gamma_3 C_{l_{\delta r}} + \Gamma_4 C_{n_{\delta r}} \\ C_{r_0} &= \Gamma_4 C_{l_0} + \Gamma_8 C_{n_0} \\ C_{r_\beta} &= \Gamma_4 C_{l_\beta} + \Gamma_8 C_{n_\beta} \\ C_{r_p} &= \Gamma_4 C_{l_p} + \Gamma_8 C_{n_p} \\ C_{r_r} &= \Gamma_4 C_{l_r} + \Gamma_8 C_{n_r} \\ C_{r_{\delta a}} &= \Gamma_4 C_{l_{\delta a}} + \Gamma_8 C_{n_{\delta a}} \\ C_{r_{\delta r}} &= \Gamma_4 C_{l_{\delta r}} + \Gamma_8 C_{n_{\delta r}} \end{aligned}$$

2.3 Metode Kontrol UAV

Terdapat beragam jenis cara untuk mengontrol UAV tipe *fixed wing*. Namun, pada tugas akhir ini hanya ingin memfokuskan pada satu jenis saja yaitu LQR *tuning*-PID. Dengan menggabungkan kedua metode ini, diharapkan untuk mendapatkan semua keuntungan dari masing-masing kontroler.

2.3.1 Non-linear State Feedback Decoupler[4]

Jika dilihat dari variable yang dapat dikendalikan oleh UAV, sistem dapat disebut sebagai sistem MIMO (*Multiple Input Multiple Output*) karena memiliki banyak masukan dan banyak keluaran. Pada persamaan dinamika pesawat, persamaan matematisnya bersifat *non-linear* dan persamaan satu dengan lainnya *coupled*.

Decoupling digunakan untuk menghilangkan hubungan antar persamaan pada pesawat yang saling berinteraksi, sehingga keseluruhan

sistem MIMO dapat dirubah menjadi sistem SISO (*Single Input Single Output*). Dan agar dapat menggunakan metode LQR untuk mencari *gain* PID, diperlukan linearisasi. Proses linearisasi digunakan dengan cara mengumpukan balik *state* dari UAV sehingga proses decoupling memaksa untuk menghilangkan interaksi antara sub-sistem dan melinearkan persamaan dinamika pesawat. Karena variable yang dokontrol adalah sudut defleksi *aileron*, *elevator*, dan *rudder*, maka persamaan yang dilinearisasi ialah persamaan dinamika lateral dari pesawat. Di dalam *state space*. Suatu sistem *non-linear* dapat dinyatakan dengan Persamaan (2.35) dan Persamaan (2.36).

$$\dot{\mathbf{x}}(t) = \mathbf{G}(\mathbf{x}, t) + \mathbf{F}(\mathbf{x}, t)\mathbf{u} \quad (2.35)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (2.36)$$

di mana,

$\mathbf{G}(\mathbf{x}, t)$ = Matriks *state* sistem *non-linear* dan *decoupled*

$\mathbf{F}(\mathbf{x}, t)$ = Matriks input yang memiliki invers atau pseudoinverse

\mathbf{C} = Matriks output

Bila terdapat suatu matriks sistem \mathbf{A} dengan nilai konstan dan *decoupled* (independen) dan matriks input \mathbf{B} yang juga bernilai konstan sehingga dibentuk persamaan *state* baru yaitu

$$\dot{\mathbf{x}}(t) = \mathbf{G}(\mathbf{x}, t) + \mathbf{F}(\mathbf{x}, t)\mathbf{u} + \mathbf{Ax} - \mathbf{Ax} + \mathbf{B}\hat{\mathbf{u}} - \mathbf{B}\hat{\mathbf{u}} \quad (2.37)$$

di mana,

$\hat{\mathbf{u}}$ = input baru

Setelah itu persamaan dapat disusun ulang menjadi

$$\dot{\mathbf{x}}(t) = \mathbf{Ax} + \mathbf{B}\hat{\mathbf{u}} + \mathbf{G}(\mathbf{x}, t) + \mathbf{F}(\mathbf{x}, t)\mathbf{u} - \mathbf{Ax} - \mathbf{B}\hat{\mathbf{u}} \quad (2.38)$$

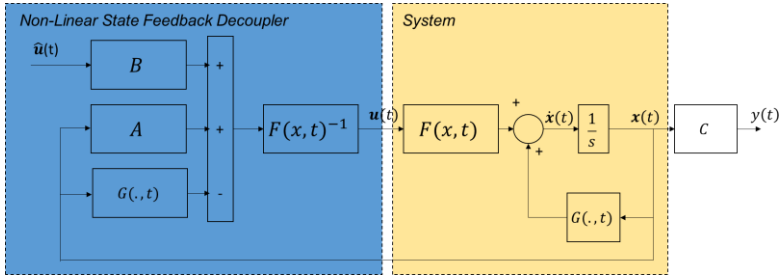
Namun, agar dapat menjadi linear dan *decoupled*, sistem harus menjadi

$$\dot{\mathbf{x}}(t) = \mathbf{Ax} + \mathbf{B}\hat{\mathbf{u}} + \underbrace{\mathbf{G}(\mathbf{x}, t) + \mathbf{F}(\mathbf{x}, t)\mathbf{u} - \mathbf{Ax} - \mathbf{B}\hat{\mathbf{u}}}_0 \quad (2.39)$$

Nilai 0 dapat dicapai dengan merubah nilai \mathbf{u} dengan syarat matriks $\mathbf{F}(\mathbf{x}, t)$ harus memiliki invers atau pseudoinverse. Persamaan baru \mathbf{u} agar Persamaan (2.39) menjadi linier dan *decoupled* ialah

$$\mathbf{u} = \mathbf{F}(\mathbf{x}, t)^{-1}(\mathbf{Ax} + \mathbf{B}\hat{\mathbf{u}} - \mathbf{G}(\mathbf{x}, t)) \quad (2.40)$$

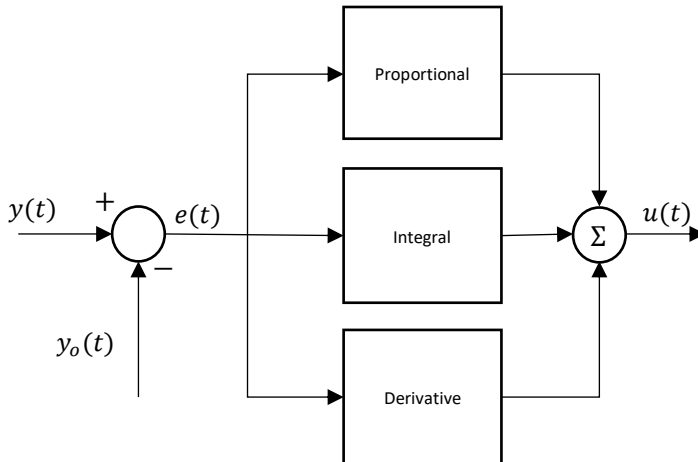
Sistem ini dapat digambarkan oleh Gambar 2.3 berikut



Gambar 2.3 Diagram blok *Non-Linear State Feedback Decoupler*[6]

2.3.2 Pengendali PID

Kontroler PID merupakan gabungan dari tiga komponen yaitu komponen proporsional (K_p), integral (K_i), dan derivatif (K_d) yang sederhana dan cukup efektif dalam implementasinya. Setiap komponen dalam kontroler PID memiliki tugas masing – masing dalam mengendalikan UAV. Arsitektur kontroler PID dapat dilihat dalam Gambar 2.4



Gambar 2.4 Arsitektur pengendali PID

Untuk kalkulasi yang terdapat di dalam pengendali ini ialah

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (2.41)$$

di mana,

- $u(t)$ = Sinyal kontrol
- K_p = *Gain* proporsional
- K_i = *Gain* integral
- K_d = *Gain* derivatif
- $e(t)$ = *Error steady state*

Setiap komponen dari kontroler PID juga memiliki karakteristik tersendiri yang dapat digunakan untuk *tuning gain*. Dengan menaikkan nilai *proportional gain* (K_p), akan memberikan *rise time* yang lebih cepat namun tingkat *overshoot* juga akan meningkat, serta mengurangi nilai *steady state error*. Dengan menambah *integral gain* (K_i), juga akan mempercepat *rise time* dan meningkatkan *overshoot*. Namun, penambahan integral gain dapat meningkatkan tingkat *settling time* dan mengurangi *steady state error*. Penambahan *derivative gain* (K_d) akan mengurangi *overshoot* dan dapat membuat respon sistem menjadi lebih halus. Secara keseluruhan karakteristik dari masing-masing gain dapat dilihat dari Tabel 2.1

Tabel 2.1 Karakteristik komponen dalam PID[9]

CL Responses	Rise Time	Overshoot	Settling Time	S-S Error
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Decrease
Kd	Small Change	Decrease	Decrease	No Change

2.3.3 Linear Quadratic Regulator[2]

Linear Quadratic Regulator atau LQR merupakan salah satu tipe dari kontrol optimal yang memfokuskan dalam meminimalisir *cost* dari dinamika sistem. Indeks performansi dari LQR ialah merupakan persamaan kuadrat seperti berikut

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (2.42)$$

Di mana, Q dan R merupakan *weighted matrices* dari *vector state* dan *vector input*. Pada indeks performansi, determinan matriks Q

menentukan seberapa kecil kesalahan dari sistem. Nilai determinan Q yang besar akan membuat kesalahan keluaran dari sistem sekecil mungkin, Sedangkan determinan matriks R menentukan seberapa kecil penggunaan *energy drive* dari sistem. Semakin besar nilai determinan R , penggunaan *energy drive* akan menjadi kecil. Sehingga dengan indeks performansi dari Persamaan (2.42), akan dibuat sinyal kendali dengan nilai kesalahan keluaran yang minimum, namun di sisi lain diinginkan juga penggunaan *energy drive* yang kecil.

Tujuan dari kontrol optimal ialah untuk menghitung *feedback control law*, yang menjamin *cost function* akan memiliki nilai minimum. Hukum kontrol optimal dapat ditulis dengan

$$u = -K_c x \quad (2.43)$$

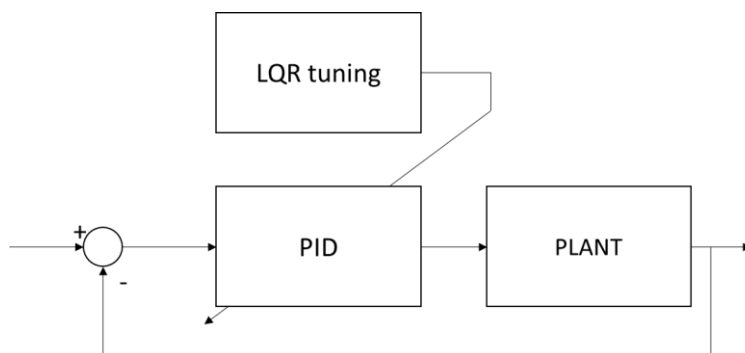
$$K_c = R^{-1} B^T X \quad (2.44)$$

Dan P merupakan solusi dari persamaan Riccati dengan persamaan

$$XA + A^T X - XBR^{-1}BX + Q = 0 \quad (2.45)$$

2.3.4 Pengendali LQR *tuning* – PID

Dalam strategi untuk mencari *gain* proporsional, integral, dan derivatif dari kontroler PID. Diantaranya terdapat metode LQR *tuning* – PID. Yaitu, dengan memanfaatkan persamaan Riccati dari metode LQR untuk mendapatkan nilai *gain* proporsional, integral, dan derivatif. Diagram blok pengendali ini dapat dilihat pada Gambar 2.5.



Gambar 2.5 Diagram blok LQR *tuning* – PID

Hal ini dilakukan untuk menghindari harga tinggi dan nilai kompleksitas tinggi dalam merealisasikan kontrol regulator optimal sederhana. Jika didapatkan bentuk *steady-state* dari suatu sistem

$$\dot{x} = Ax + Bu \quad \text{dengan } x(0) = x_0 \quad (2.46)$$

$$y = Cx \quad (2.47)$$

Di mana x merupakan n -vector *state* dari sistem, u adalah m -vector *input* ke sistem, y adalah l -vektor *output* dari sistem dan A, B, C adalah matriks konstan dengan dimensi yang cocok.

Hal ini dapat didemonstrasikan saat mencoba untuk mencari gain pada sistem orde 2. Contoh pada persamaan berikut

$$\frac{Y(s)}{U(s)} = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} \quad (2.48)$$

Dalam persamaan differensial, model dapat dinyatakan sebagai berikut

$$\ddot{s} + a_1 \dot{s} + a_0 = b_1 \dot{u} + b_0 u \quad (2.49)$$

Jika dimisalkan $x_1 = y$ dan $x_2 = \dot{y} - b_1 u$, maka hasil dari persamaan *state* bisa didapat

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -a_0 & -a_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_0 - a_1 b_1 \end{pmatrix} u \quad (2.50)$$

Lalu ditambahkan *state* baru

$$x_3 = \int x_1 dt \rightarrow \dot{x}_3 = x_1 \quad (2.51)$$

Persamaan *state* baru menjadi

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -a_0 & -a_1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_0 - a_1 b_1 \\ 0 \end{pmatrix} u \quad (2.52)$$

$$y = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Persamaan *state* diatas memiliki matriks *augmented* untuk A, B, C . Untuk diubah menjadi PID *equivalent*, digunakan persamaan Riccati untuk mendapatkan X . Dan nilai *gain* bisa didapatkan dengan Persamaan (2.53) berikut.

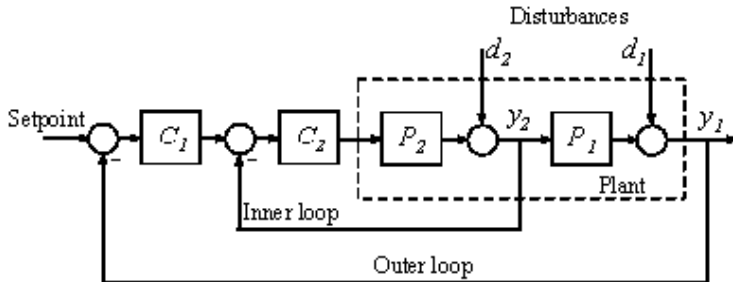
$$K = R^{-1} B^T X \quad (2.53)$$

di mana,

$$K = [K_p \ K_d \ K_i]$$

2.3.5 Pengendali Ketinggian dengan PID Cascade[7]

Konfigurasi *cascade* dilakukan dengan membuat dua loop yaitu *inner loop* untuk *secondary controller* dan *outer loop* untuk *primary controller*. Di dalam kasus ini, *outer loop* memiliki referensi ketinggian yang ingin dicapai oleh pesawat. Dan *inner loop* digunakan untuk mengontrol sudut *pitch* dari pesawat. Secara teori, konfigurasi *cascade* memiliki blok diagram sebagai berikut.



Gambar 2.6 Teori Konfigurasi *Cascade*[7]

Di mana C_1 merupakan *primary controller* pada *outer loop* yang nantinya akan digunakan untuk mengendalikan ketinggian dan C_2 merupakan *secondary controller* yang nantinya akan digunakan untuk mengendalikan sudut *heading pitch* dari pesawat.

2.4 Path Following

Path following dapat dikatakan sebagai kemampuan sebuah kendaraan otomatis dalam mengikuti sebuah referensi berupa *waypoint* atau jalur. Sistem ini sering digunakan sebagai salah satu cara agar kendaraan bisa berjalan tanpa awak, dalam laporan ini berupa UAV, dalam sebuah *space* tertentu. Sistem ini memerlukan sebuah *waypoint planner*. Terdapat dua komponen dalam sistem ini yang digunakan yang terdiri dari *Path Generator* dan *Guidance*.

2.4.1 Path Generator

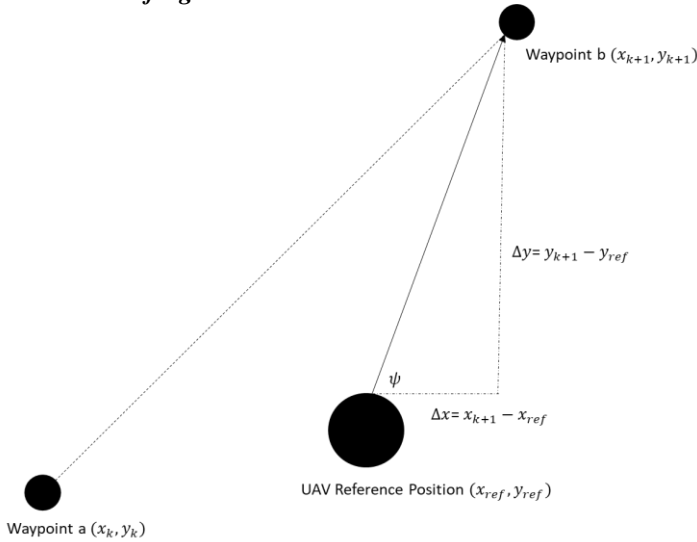
Path generator adalah merupakan sebuah matriks $n \times 2$ yang setiap komponen dalam matriksnya mewakili posisi *waypoint* pada sumbu - x

dan sumbu - y dari bumi. Kolom pertama mewakili sumbu - x dan kolom kedua mewakili sumbu - y.

2.4.2 Guidance[5]

Merupakan metode untuk menerjemahkan posisi *waypoint* dari *path generator* dan posisi UAV menjadi referensi sudut yang digunakan sebagai input pengendali PID. Terdapat dua teori utama dari *guidance* yang digunakan dalam tugas akhir ini. Yaitu.

2.4.2.1 Line of sight



Gambar 2.7 *Line of Sight*

Diberikan *waypoint* $p_k = [x_k \ y_k]^T \in \mathbb{R}^2$ dan $p_{k+1} = [x_{k+1} \ y_{k+1}]^T \in \mathbb{R}^2$ dengan sebuah garis lurus diantaranya seperti pada gambar. Tujuannya ialah agar UAV mengikuti garis tersebut dengan *cross-track error* sekecil mungkin.

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (2.54)$$

Dengan persamaan *cross-track error* sebagai berikut

$$e(t) = -[x(t) - x_k] \sin(\alpha_k) + [y(t) - y_k] \cos(\alpha_k) \quad (2.55)$$

Di mana sudut α_k yang digunakan untuk memutar *north-axis* di dalam *earth frame* dengan titik *origin* p_k^n ditulis sebagai

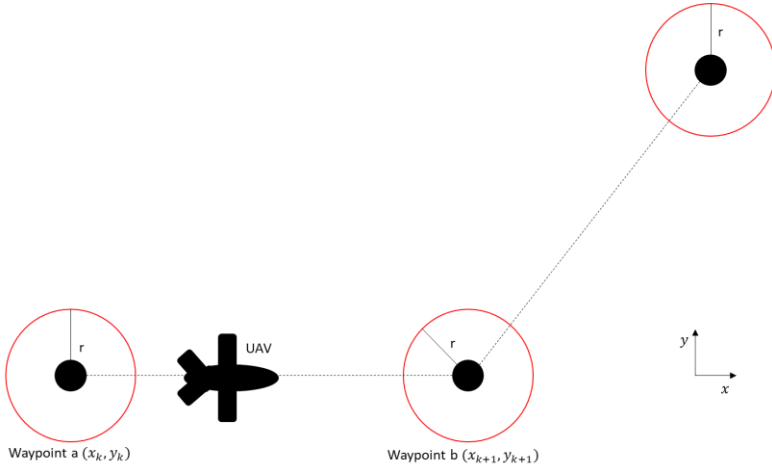
$$\alpha_k := \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k) \in S := [-\pi, \pi] \quad (2.56)$$

2.4.2.2 Circle of acceptance

Circle of acceptance adalah sebuah metode agar mengetahui kapan untuk berpindah ke *waypoint* berikutnya, dari $[x_k \ y_k]^T \in R^2$ ke $[x_{k+1} \ y_{k+1}]^T \in R^2$, di mana *x* adalah *north* dan *y* adalah *east* dari frame NED. Ketika pesawat berada di dalam lingkaran dengan radius $R_{k+1} \in R^1$ mengelilingi point $[x_{k+1} \ y_{k+1}]^T$, seperti pada gambar di bawah, *guidance system* akan berganti ke *waypoint* berikutnya. Posisi UAV harus memenuhi persamaan berikut

$$[x_{k+1} - x(t)]^2 + [y_{k+1} - y(t)]^2 \leq R_{k+1}^2 \quad (2.57)$$

pada saat *t* untuk mengganti *waypoint*. Gambar 2.8 menunjukkan hukum *circle of acceptance* di dalam 2-dimensional plane.



Gambar 2.8 *Circle of Acceptance*

BAB 3

PERANCANGAN SISTEM

Bab ini menjelaskan tahap – tahap pembuatan simulasi dari sistem *path following* UAV. Pertama yaitu membuat model dinamika pesawat, dan perancangan model linearisasi pesawat dengan *decoupler*, lalu dilanjutkan dengan perancangan kontroler *LQR tuning*-PID, serta perancangan *waypoint planner* untuk sistem *path following* dari UAV.

3.1 Gambaran Umum Sistem

Terdapat 5 tahap yang perlu diperhatikan dalam merancang sismulasi *path following*. Pertama – tama harus dirancang model matematika pesawat. Selanjutnya dari model matematika yang ada, dapat merancang kontroler untuk mengatur output dari plant. Penelitian pada buku tugas akhir ini menggunakan pengendali *LQR tuning* – PID untuk mengendalikan output sudut *roll*, *pitch*, dan *yaw* dari pesawat. Agar pengendalian dapat dilakukan, digunakan *decoupler* untuk menghilangkan interaksi antar persamaan dinamika rotasi pesawat. Lalu, diperlukan kendali ketinggian agar dapat mengatur ketinggian dari pesawat.

Pesawat akan diuji dari dua jenis referensi yaitu, referensi berupa nilai konstan, dan referensi berupa jalur. Pesawat dapat dikendalikan untuk mengikuti jalur dengan strategi mengatur sudut yang mengakibatkan berubahnya arah *heading* dari pesawat.

3.2 Perancangan Model UAV *Fixed Wing*

Perancangan model matematika model UAV tipe *fixed wing* merupakan tahap pertama dari seluruh perancangan sistem. Hukum Newton II menjadi acuan yang digunakan untuk merancang sistem. Hukum tersebut menyatakan bahwa percepatan suatu benda berbanding lurus dengan jumpah gaya yang bekerja pada benda tersebut, dan berbanding terbalik dengan massanya. Dari hukum ini dapat dirancang sebuah model matematika UAV seperti berikut

$$\dot{u} = rv - qw - g \sin \theta + \frac{\rho V_a^2 S}{2m} \left[C_X(\alpha) + C_{X_q}(\alpha) \frac{cq}{2V_a} + C_{X_{\delta e}}(\alpha) \delta_e \right] + \frac{\rho S_{prop} c_{prop}}{2m} [(k_{motor} \delta_t)^2 - V_a^2] \quad (3.1a)$$

$$\dot{v} = pw - ru - g \cos \theta \sin \phi + \frac{\rho V_a^2 S}{2m} \left[C_{Y_0} + C_{Y_\beta} \beta + C_{Y_p} \frac{bp}{2V_a} + C_{Y_r} \frac{br}{2V_a} + C_{Y_{\delta a}} \delta_a + C_{Y_{\delta r}} \delta_r \right] \quad (3.1b)$$

$$\dot{w} = qu - pv - g \cos \theta \cos \phi + \frac{\rho V_a^2 S}{2m} \left[C_Z(\alpha) + C_{Z_q}(\alpha) \frac{cq}{2V_a} + C_{Z_{\delta e}}(\alpha) \delta_e \right] \quad (3.1c)$$

$$\dot{p} = \Gamma_1 pq - \Gamma_2 qr + \frac{\rho V_a^2 S b}{2} \left[C_{p_0} + C_{p_\beta} \beta + C_{p_p} \frac{bp}{2V_a} + C_{p_r} \frac{br}{2V_a} + C_{p_{\delta a}} \delta_a + C_{p_{\delta r}} \delta_r \right] \quad (3.1d)$$

$$\dot{q} = \Gamma_5 pq - \Gamma_6 (p^2 - r^2) + \frac{\rho V_a^2 S c}{2I_y} \left[C_{m_0} + C_{m_a} + C_{m_q} \frac{cq}{2V_a} + C_{m_{\delta e}} \delta_e \right] \quad (3.1e)$$

$$\dot{r} = \Gamma_7 pq - \Gamma_8 qr + \frac{\rho V_a^2 S b}{2} \left[C_{r_0} + C_{r_\beta} \beta + C_{r_p} \frac{bp}{2V_a} + C_{r_r} \frac{br}{2V_a} + C_{r_{\delta a}} \delta_a + C_{r_{\delta r}} \delta_r \right] \quad (3.1f)$$

Persamaan dinamika dari dua jenis yaitu dinamika longitudinal dan dinamika lateral. Kedua jenis persamaan dinamika merupakan persamaan yang menunjukkan gaya – gaya yang bekerja pada pesawat. Gaya – gaya yang bekerja pada pesawat antara lain gaya aerodinamika, gravitasi, gaya *lift*, gaya *drag*, dan berbagai macam lainnya. Dengan persamaan dinamika dijabarkan pada Persamaan (3.1a) hingga Persamaan (3.1f).

Model dibuat dengan menggunakan fitur Simulink dalam perangkat lunak MATLAB. Blok tersebut adalah MATLAB Function dari kategori User-Defined Function. Blok ini memiliki isi berupa parameter dan kumpulan persamaan non-linear serta transformasi koordinat dari body frame ke earth frame. Transformasi matriks dapat dijabarkan sebagai berikut

$$C = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\sin\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ \sin\theta & -\sin\phi\cos\theta & -\cos\phi\cos\theta \end{bmatrix}$$

$$\begin{bmatrix} P_{north} \\ P_{east} \\ h \end{bmatrix} = C \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (3.2a)$$

$$D = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix}$$

$$\begin{bmatrix} \phi_{earth} \\ \theta_{earth} \\ \psi_{earth} \end{bmatrix} = D \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.2b)$$

Di mana P_{north} merupakan sumbu-x pada terhadap bumi, P_{east} merupakan sumbu-y pada bumi, dan h merupakan ketinggian atau sumbu-z terhadap bumi. Demikian juga dengan ϕ_{earth} , θ_{earth} , dan ψ_{earth} merupakan hasil transformasi *roll*, *pitch*, *yaw* terhadap bumi.

Blok ini memiliki keluaran berupa percepatan translasi dan percepatan rotasi dari pesawat berdasarkan body frame, serta perubahannya ke earth frame. Dengan memberikan banyak integrator setelah keluaran tersebut, dapat diambil data berupa sudut *roll*, *pitch*, dan *yaw* pada body frame. Data tersebut akan diumpan balikan ke pengendali PID untuk mengontrol sudut.

Pada persamaan dinamika UAV, nilai – nilai dari parameter didapatkan dari [1] yang dapat langsung dimasukkan ka dalam blok Simulink.

Tabel 3.1 Parameter UAV[1]

Parameter	Keterangan
α	<i>Angle of attack</i>
β	<i>Side Slip angle</i>
J_x	Inersia pada sumbu-x
J_y	Inersia pada sumbu-y
J_z	Inersia pada sumbu-z
J_{xz}	Inersia pada sumbu dibentang sumbu x dan z
m	Massa total UAV
g	Grafitasi
Γ	Produk dari Matrix Inersia

Parameter	Keterangan
V_a	Vektor <i>Airspeed</i>
V_g	Vektor kecepatan <i>ground</i>
V_w	Vektor kecepatan Angin
C_L	Koefisien angkat (fixed-wing)
C_D	Koefisien drag (fixed-wing)
C_m	Koefisien momen <i>pitching</i>
C_l	Koefisien momen <i>rolling</i>
C_n	Koefisien momen yawing
C_X	Koefisien gaya pada X_B
C_Y	Koefisien gaya pada Y_B
C_Z	Koefisien gaya pada Z_B
k_{motor}	konstanta efisiensi motor
S_{prop}	luas yang dihempas propeller
C_{prop}	Koefisien aerodinamis propeller
S	Area penampang sayap pesawat
c	<i>Mean cord</i> sayap
b	Bentang sayap
ρ	Densitas atmosfer

Tabel 3.2 Estimasi parameter UAV[1]

Simbol	Nilai
J_x	0.1147 (Kg m ²)
J_y	0.0576 (Kg m ²)
J_z	0.1712 (Kg m ²)
J_{xz}	0.0015 (Kg m ²)
K_p	1.3678x10 ⁻⁵ (N s ²)
K_d	2.4323x10 ⁻⁵ (N m s ²)
m	1.56 (Kg)
g	9.81 (m/s ²)
$C_{L.}$	0.09167 (0), 3.5016 (α), 2.8932 (q), 0.2724 (δ_e)
$C_{D.}$	0.09167 (0), 3.5016 (α), 2.8932 (q), 0.2724 (δ_e)
$C_{m.}$	-0.02338 (0), -0.5675 (α), -1.399 (q), -0.3254 (δ_e)
b	2.8956
$C_{l.}$	0 (0), -0.02854 (β), -0.3209 (p), - 0.3209 (r), 0.1682 (δ_a), 0.105 (δ_r)
c	0.3302 m

Simbol	Nilai
C_n	0 (0), -0.0004 (β), -0.01297 (p), -0.00434 (r), -0.00328 (δ_a), -0.032 (δ_r)
ρ	1.2682 (kg/m ³)
C_Y	0 (0), -0.07359 (β), 0 (p), 0 (r), 0 (δ_e)
k_{motor}	20
S_{prop}	0.0314 m ²
C_{prop}	1
S	0.2589 m ²

3.3 Perancangan *Non-Linear State Feedback Decoupler*

Sebelum mengolah data pada metode LQR *tuning*, sebelumnya harus dilakukan linearisasi. Untuk merancang *Non-linear State Feedback Decoupler* persamaan dinamika yang difokuskan ialah persamaan lateral. Karena variabel yang dikendalikan adalah sudut orientasi dari pesawat. Jika dilihat dari Persamaan (3.1a) hingga (3.1f) terlihat bahwa persamaan dinamika pesawat bersifat *non-linear* dan *coupled*. Sistem harus dilinearisasi agar persamaan dinamika menjadi linear. Dan *decoupler* agar persamaan dinamika menjadi *decoupled* (interaksi antar persamaan hilang).

Linearisasi dilakukan dengan menghilangkan momen – momen non-linear yang bekerja pada pesawat secara paksa. Input dari persamaan dinamika sudut defleksi dari *aileron*, *elevator*, dan *rudder* δ_a , δ_e , dan δ_r . Parameter baru dengan *time constant* (τ) sesuai keinginan ditambahkan agar dinamika rotasi pesawat memiliki respon seperti dengan sistem orde satu. *Time constant* yang dipilih untuk tugas akhir ini ialah sebesar 0.1 detik. Lalu, parameter yang baru ditambahkan agar pesawat tetap dapat dikendalikan. Maka, persamaan baru untuk input dinamika pesawat adalah sebagai berikut

$$C_{p\delta_a}\delta_a + C_{p\delta_r}\delta_r = \frac{2}{\rho V_a^2 S b} \left(-\Gamma_1 p q + \Gamma_2 q r - \frac{\rho V_a^2 S b}{2} \left(C_{p_0} + C_{p_\beta} \beta + C_{p_p} \frac{bp}{2V_a} + C_{p_r} \frac{br}{2V_a} \right) - \lambda_1 p + u_1 \right) \quad (3.3a)$$

$$\delta_e = \frac{2m}{\rho V_a^2 S C_{m_{\delta e}}} \left(-\Gamma_5 p q + \Gamma_6 (p^2 - r^2) - \frac{\rho V_a^2 S c}{2 l_y} \left(C_{m_0} + C_{m_a} + C_{m_q} \frac{c q}{2 V_a} \right) - \lambda_2 q + u_2 \right) \quad (3.3b)$$

$$C_{r_{\delta a}} \delta_a + C_{r_{\delta r}} \delta_r = \frac{2}{\rho V_a^2 S b} \left(-\Gamma_7 p q + \Gamma q r - \frac{\rho V_a^2 S b}{2} \left(C_{r_0} + C_{r_\beta} \beta + C_{r_p} \frac{b p}{2 V_a} + C_{r_r} \frac{b r}{2 V_a} \right) - \lambda_3 r + u_3 \right) \quad (3.3c)$$

Untuk mempermudah penyelesaian, Persamaan (3.3a), (3.3b), dan (3.3c) bisa dirubah dalam bentuk matriks.

$$\begin{bmatrix} C_{p_{\delta a}} & 0 & C_{p_{\delta r}} \\ 0 & C_{m_{\delta e}} & 0 \\ C_{r_{\delta a}} & 0 & C_{r_{\delta r}} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} = \begin{bmatrix} \frac{2}{\rho V_a^2 S b} \left(-\Gamma_1 p q + \Gamma_2 q r - \frac{\rho V_a^2 S b}{2} \left(C_{p_0} + C_{p_\beta} \beta + C_{p_p} \frac{b p}{2 V_a} + C_{p_r} \frac{b r}{2 V_a} \right) - \lambda_1 p + u_1 \right) \\ \frac{2m}{\rho V_a^2 S} \left(-\Gamma_5 p q + \Gamma_6 (p^2 - r^2) - \frac{\rho V_a^2 S c}{2 l_y} \left(C_{m_0} + C_{m_a} + C_{m_q} \frac{c q}{2 V_a} \right) - \lambda_2 q + u_2 \right) \\ \frac{2}{\rho V_a^2 S b} \left(-\Gamma_7 p q + \Gamma q r - \frac{\rho V_a^2 S b}{2} \left(C_{r_0} + C_{r_\beta} \beta + C_{r_p} \frac{b p}{2 V_a} + C_{r_r} \frac{b r}{2 V_a} \right) - \lambda_3 r + u_3 \right) \end{bmatrix} \quad (3.4)$$

Jika dimisalkan,

$$D = \frac{2}{\rho V_a^2 S b} \left(-\Gamma_1 p q + \Gamma_2 q r - \frac{\rho V_a^2 S b}{2} \left(C_{p_0} + C_{p_\beta} \beta + C_{p_p} \frac{b p}{2 V_a} + C_{p_r} \frac{b r}{2 V_a} \right) - \lambda_1 p + u_1 \right) \quad (3.5a)$$

$$E = \frac{2m}{\rho V_a^2 S} \left(-\Gamma_5 p q + \Gamma_6 (p^2 - r^2) - \frac{\rho V_a^2 S c}{2 l_y} \left(C_{m_0} + C_{m_a} + C_{m_q} \frac{c q}{2 V_a} \right) - \lambda_2 q + u_2 \right) \quad (3.5b)$$

$$F = \frac{2}{\rho V_a^2 S b} \left(-\Gamma_7 p q + \Gamma q r - \frac{\rho V_a^2 S b}{2} \left(C_{r_0} + C_{r_\beta} \beta + C_{r_p} \frac{b p}{2 V_a} + C_{r_r} \frac{b r}{2 V_a} \right) - \lambda_3 r + u_3 \right) \quad (3.5c)$$

$$L = \begin{bmatrix} C_{p_{\delta a}} & 0 & C_{p_{\delta r}} \\ 0 & C_{m_{\delta e}} & 0 \\ C_{r_{\delta a}} & 0 & C_{r_{\delta r}} \end{bmatrix} \quad (3.5d)$$

Dengan mensubstitusi Persamaan (3.5a) hingga (3.5d) ke Persamaan (3.4) maka akan didapatkan

$$L \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} = \begin{bmatrix} D \\ E \\ F \end{bmatrix} \quad (3.6)$$

Untuk mendapatkan nilai input bisa dilakukan pemindahan matriks sederhana yaitu

$$\begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} = L^{-1} \begin{bmatrix} D \\ E \\ F \end{bmatrix} \quad (3.7)$$

dengan,

$$\lambda_1 = \frac{1}{\tau_p} = 10$$

$$\lambda_2 = \frac{1}{\tau_q} = 10$$

$$\lambda_3 = \frac{1}{\tau_r} = 10$$

Pada perancangan simulasi, blok model dinamika tetap dibiarkan berbentuk *non-linear*. Proses linearisasi dan *decoupling* dilakukan diluar dari model dinamika UAV. Blok linearisasi dan *decoupler* dirancang dengan menggunakan blok MATLAB *Function* yang terdapat di Simulink. Blok ini memiliki *input* sudut orientasi, serta *feedback* dari blok integrator.

3.4 Perancangan Kontroler LQR *tuning* – PID

Dalam sistem pengendalinya, secara sederhana, LQR tuning-PID bisa didefinisikan sebagai pengendali PID dengan nilai gain yang didapatkan dari metode LQR yaitu dengan cara merubah – ubah nilai matriks Q dan R agar mendapatkan nilai yang sesuai dengan spesifikasi yang diinginkan. Metode ini dilakukan dengan harapan agar pengendali LQR dapat diterjemahkan kedalam pengendali PID. Pada perancangan pengendali untuk *roll*, *pitch*, dan *yaw* memiliki karakteristik dasar yang sama yaitu memiliki nilai $\lambda = 10$. Namun, hal yang berbeda ialah nilai matriks Q dan R pada masing – masing pengendali. Proses perancangannya ialah sebagai berikut.

Roll

$$\dot{\phi} = p \quad (3.8a)$$

$$\dot{p} = -\lambda_1 p + u_1 \quad (3.8b)$$

Dengan memisalkan $\phi = y$ didapatkan persamaan *state*

$$\begin{bmatrix} \dot{\phi} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\lambda_1 \end{bmatrix} \begin{bmatrix} \phi \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_1 \quad (3.8c)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ p \end{bmatrix} \quad (3.8d)$$

Matriks augmentasi bisa didapatkan dengan cara seperti berikut[3]

$$\bar{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \text{ \& } \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

Yang menyebabkan hasil dari matriks augmentasi

$$\bar{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\lambda_1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \text{ \& } \bar{B} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (3.8e)$$

Pitch

$$\dot{\theta} = q \quad (3.9a)$$

$$\dot{q} = -\lambda_2 q + u_2 \quad (3.9b)$$

Dengan memisalkan $\theta = y$ didapatkan persamaan *state*

$$\begin{bmatrix} \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\lambda_2 \end{bmatrix} \begin{bmatrix} \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_2 \quad (3.9c)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ q \end{bmatrix} \quad (3.9d)$$

Lalu nilai matriks augmentasinya adalah sebagai berikut

$$\bar{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\lambda_2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \text{ \& } \bar{B} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (3.9e)$$

Yaw

$$\dot{\psi} = r \quad (3.10a)$$

$$\dot{r} = -\lambda_3 r + u_3 \quad (3.10b)$$

Dengan memisalkan $\psi = y$ didapatkan persamaan *state*

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\lambda_3 \end{bmatrix} \begin{bmatrix} \psi \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_3 \quad (3.10c)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ r \end{bmatrix} \quad (3.10d)$$

Lalu nilai matriks augmentasinya adalah sebagai berikut

$$\bar{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\lambda_3 & 0 \\ 1 & 0 & 0 \end{bmatrix} \text{ \& } \bar{B} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (3.10e)$$

Setelah semua *state* didapatkan beserta matriks augmentasinya, ditentukan nilai matriks Q dan R untuk masing masing pengendali. Pada kasus ini digunakan seperti berikut

Roll

$$Q = \begin{bmatrix} 20000 & 0 & 0 \\ 0 & 300 & 0 \\ 0 & 0 & 2000 \end{bmatrix} \& R = 0,01 \quad (3.11)$$

Pitch

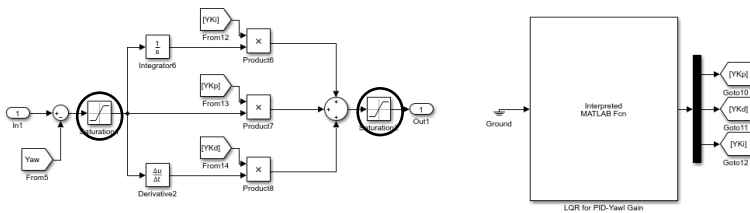
$$Q = \begin{bmatrix} 80000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 10 \end{bmatrix} \& R = 0,01 \quad (3.12)$$

Yaw

$$Q = \begin{bmatrix} 50000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 10 \end{bmatrix} \& R = 0,01 \quad (3.13)$$

Untuk simulasi, digunakan blok *Interpreted MATLAB Function* untuk memberikan nilai gain ke pengendali PID.

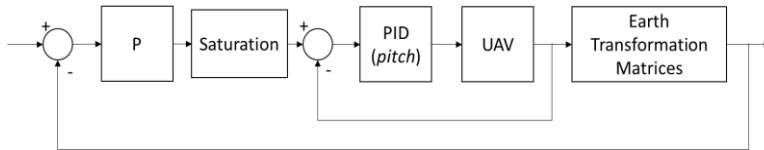
Namun dalam sistem *path following*, sebelum pengujian dilakukan, terdapat beberapa hal pada komponen – komponen yang ada di Simulink yang harus diubah sedikit. Yaitu dengan menambah *limit* pada integrator saat mengintegalkan percepatan sudut menjadi kecepatan sudut serta menambahkan saturasi pada kontroler yang mengendalikan respon sudut yaw. Jika dilihat dari pergerakannya, sistem *path following* ini hanya dianggap memiliki 3-DOF (*Degrees of Freedom*). Hal ini dilakukan karena ketinggian dianggap konstan selama berjalannya simulasi ini. Penambahan saturasi dapat dilihat pada Gambar 3.1 berikut.



Gambar 3.1 Penambahan saturasi pada kontroler PID untuk sudut yaw

3.5 Perancangan *Altitude Control*

Untuk mengatur ketinggian dari pesawat, digunakan konfigurasi *cascade* dengan menambahkan *outer loop* yang memiliki kontroler primer untuk mengatur ketinggian dan *inner loop* untuk mengatur *pitch* dari pesawat. Blok diagram dari pengendali ini dapat dilihat pada Gambar 3.4 berikut.



Gambar 3.2 *Altitude Control*

Di dalam blok PID pada MATLAB, tipe pengendali yang digunakan hanyalah cukup pengendali proporsional. Dengan nilai gain $K_p = 100$ bisa didapatkan nilai ketinggian yang diinginkan dengan waktu yang optimal. Blok saturasi ditambahkan agar nilai referensi yang masuk ke dalam *inner loop* tidak lebih dari 0,4 rad. Hal ini dikarenakan nilai maksimal defleksi sirip pada *elevator* tidak dapat melebihi 0,4 rad.

3.6 Perancangan *Waypoint Planner*

Pada *waypoint planner*, koordinat titik – titik terletaknya *waypoint* direpresentasikan pada matriks $n \times 2$ di mana kolom pertama merepresentasikan posisi pada sumbu-x dan kolom kedua merepresentasikan posisi pada sumbu-y. Untuk sistem *guidance* pada *waypoint planner*, digunakan dua metode *guidance* sederhana yaitu *line of sight* dan *circle of acceptance*. Sistem ini dapat ditulis pada kode yang terlampir. Blok ini memiliki *feedback* dari posisi pesawat terhadap bumi yang akan dimasukkan sebagai komputasi untuk mendapatkan arah pesawat yang harus dicapai.

BAB 4

HASIL DAN ANALISIS

Bab ini menjelaskan pengujian dari masing – masing blok yang telah dirancang dan hasil simulasi dari pesawat mengikuti referensi yang telah diberikan. Komponen yang dilakukan pengujian adalah *decoupler* pada pesawat, kontroler *LQR tuning – PID*, model pesawat ketika diberi referensi konstan, dan model pesawat ketika diberi referensi *path*.

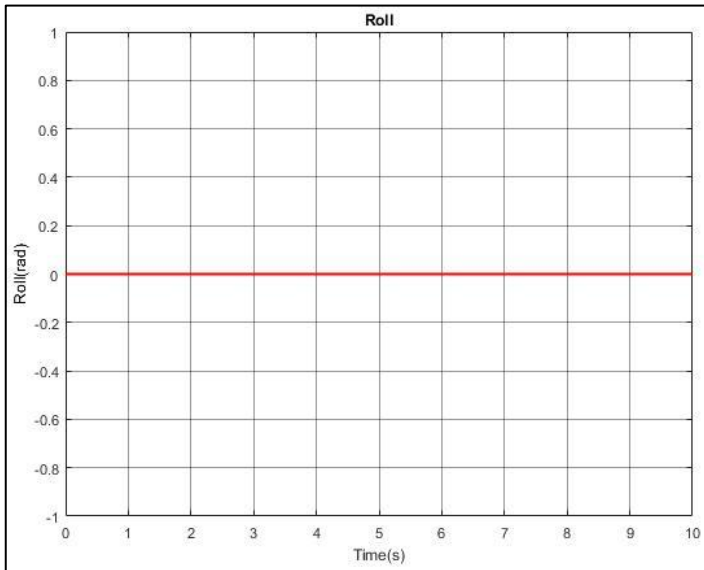
4.1 Gambaran Umum Pengujian Simulasi

Sebelum simulasi *path following* dilakukan, komponen simulasi yang telah dirancang sebelumnya harus dilakukan validasi atau pengujian terlebih dulu. Tahapan pengujian yang dilakukan adalah sebagai berikut.

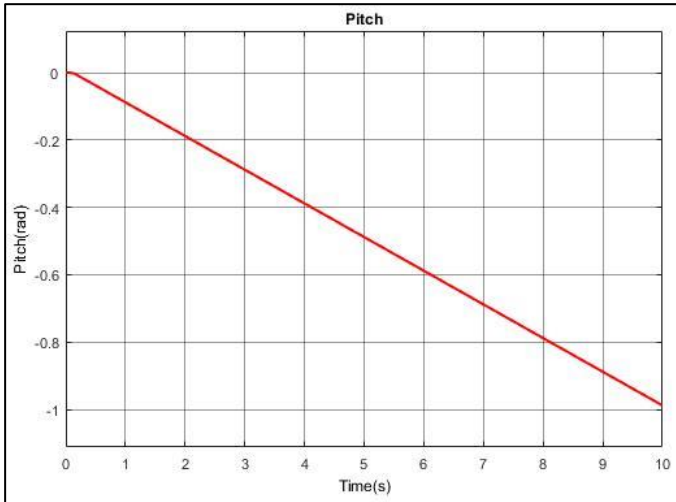
1. Pengujian model dinamika UAV
Pengujian pada model dinamika pesawat dilakukan ketika diberikan input untuk masing – masing defleksi *aileron*, *elevator*, dan *rudder* sama dengan nol dengan tanpa tambahan kontroler.
2. Pengujian *waypoint planner*
Pengujian ini dilakukan dengan memplot titik – titik *waypoint* yang harus dicapai dan menerjemahkan titik – titik tersebut menjadi referensi untuk input sistem. Lalu akan diamati referensi – referensi yang masuk kedalam sistem.
3. Pengujian pengendali *LQR tuning – PID*
Pengujian ini dilakukan untuk melihat respon sistem dengan referensi berupa konstan. Nilai referensi adalah sudut dalam satuan radian.
4. Pengujian pengendali ketinggian
Pengujian ketinggian dilakukan ketika sistem pengendali *pitch* ditambahkan *outer loop* dengan input ketinggian dan kontroler proporsional sebagai *primary controller*.
5. Pengujian sistem *path following*
Pengujian *path following* dilakukan ketika input referensi ke sistem didapatkan dari *waypoint planner* yang terdiri dari *guidance* untuk menerjemahkan titik – titik *waypoint* menjadi sudut referensi untuk sistem. Serta, dilihat grafik *cross track error* dari simulasi.

4.2 Pengujian Model Dinamika UAV

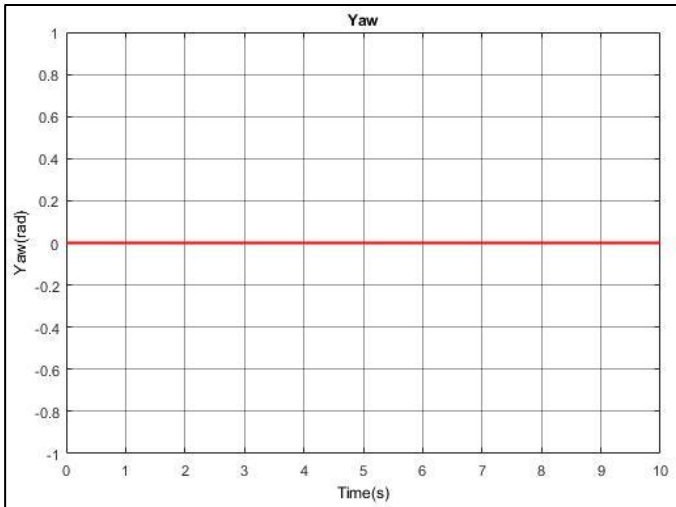
Pada model dinamika UAV, pengujian dilakukan dengan memberikan input referensi untuk defleksi bernilai nol radian untuk melihat respon sistem dari pesawat (*body frame*) serta respon pada bumi (*earth frame*). Serta kecepatan maju konstan bernilai 5m/s. Hasil simulasi pada *body frame* bisa dilihat pada Gambar 4.1, 4.2, dan 4.3. Dan untuk simulasi pada frame earth bisa dilihat pada Gambar 4.2 (a.), (b.), dan (c.).



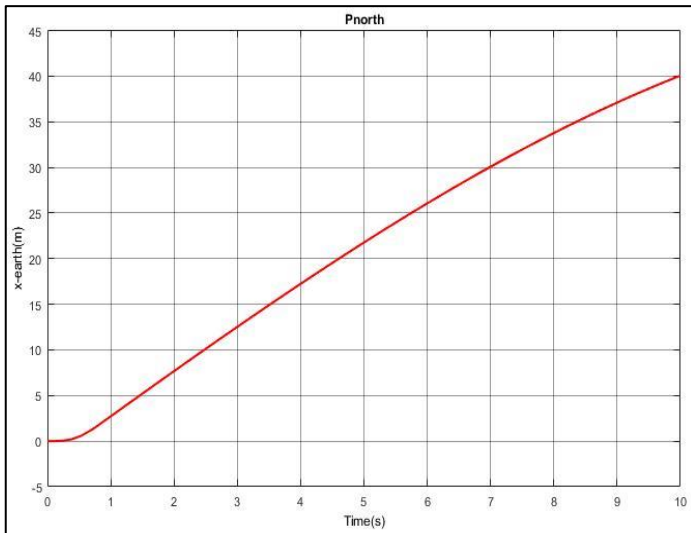
Gambar 4.1 Respon *roll* tanpa adanya pengendali



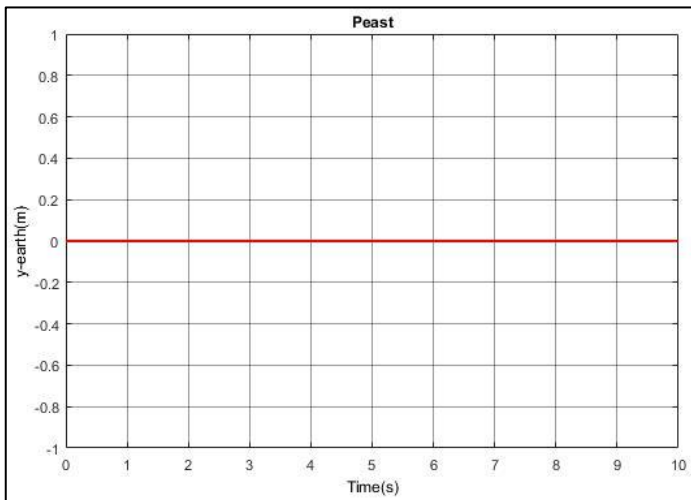
Gambar 4.2 Respon *pitch* tanpa adanya pengendali



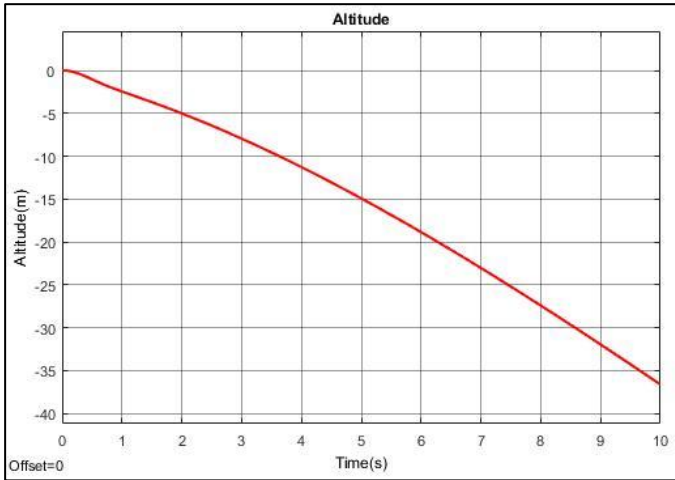
Gambar 4.3 Respon *yaw* tanpa adanya pengendali



Gambar 4.4 Posisi x terhadap bumi tanpa adanya pengendali



Gambar 4.5 Posisi y terhadap bumi tanpa adanya pengendali



Gambar 4.6 Posisi ketinggian tanpa adanya pengendali

Pada grafik simulasi pada Gambar 4.1 hingga 4.6, bisa terlihat bahwa pergerakan dari pesawat tersebut adalah gerak jatuh bebas. Hal ini dikarenakan dengan referensi yang diberikan bernilai nol. Karena dengan referensi nol, pesawat tidak akan memiliki *angle of attack* yang menyebabkan tidak ada gaya keatas untuk melawan gaya gravitasi. Maka dari itu, nilai *pitch* dan ketinggian pesawat akan terus menurun. Hal ini membuktikan bahwa untuk bergerak maju pesawat tipe *fixed wing* memerlukan *angle of attack*. Hal ini dikarenakan untuk melakukan maneuver, pesawat mengandalkan gaya aerodinamis. *Angle of attack* dapat dicari dengan rumus

$$\alpha = \tan^{-1} \left(\frac{w}{u} \right) \quad (4.1)$$

Di mana w merupakan kecepatan pada sumbu – z *body frame* dan u merupakan kecepatan pada sumbu – x *body frame*.

4.3 Pengujian Waypoint Planner

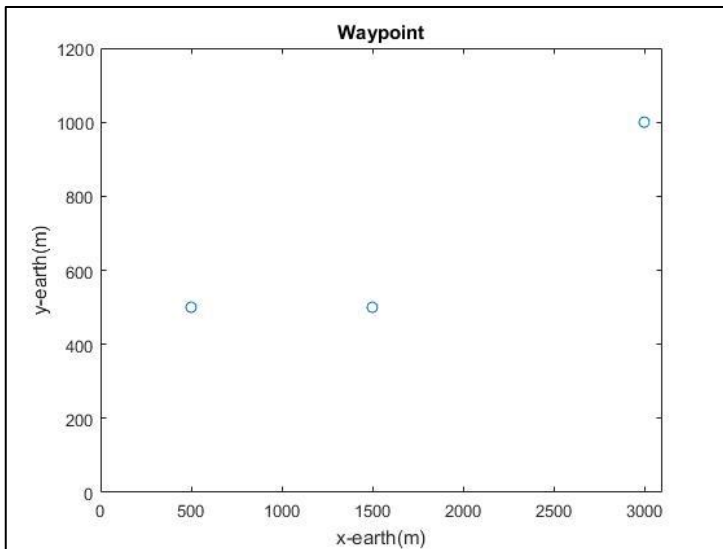
Pengujian *waypoint planner* dilakukan dengan menjalankan *script program* untuk memberikan titik – titik *waypoint* yang harus dilalui oleh pesawat. Setiap titik memiliki koordinat tertentu dalam sumbu- x dan sumbu- y dari bumi yang dapat dibentuk dari matriks $n \times 2$. Di mana, kolom

1 merepresentasikan posisi dalam sumbu-x dan kolom 2 merepresentasikan posisi dalam sumbu-y. Dalam simulasi ini, terdapat tiga titik *waypoint* yang akan digunakan. Titik- titik tersebut dapat dilihat pada Tabel 4.1.

Tabel 4.1 Titik *waypoint*

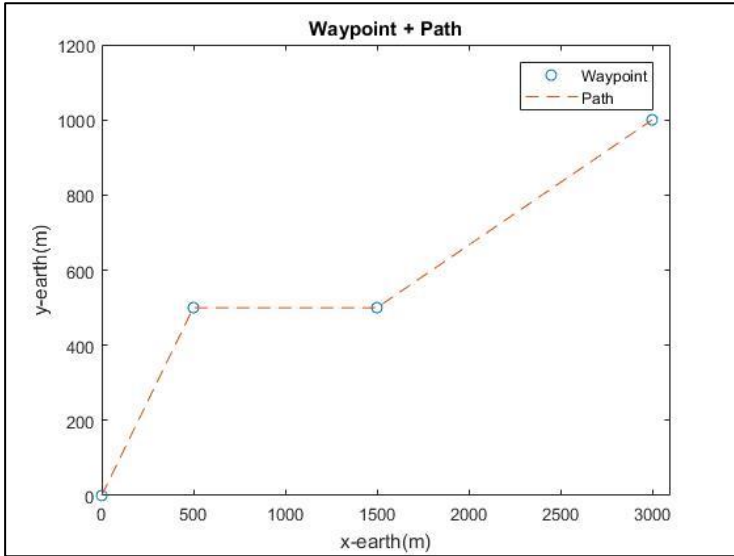
Waypoint	x	y
1	500	500
2	1500	500
3	3000	1000

Titik *waypoint* ini dapat di-plot di dalam MATLAB yang menghasilkan Gambar 4.7 berikut



Gambar 4.7 Titik – titik *waypoint*

Dikarenakan pada simulasi ini menggunakan metode *line of sight*, maka path dari pesawat merupakan hasil dari penarikan garis lurus antara *waypoint*. Inisialisasi simulasi dimulai pada titik (0,0). Maka, dapat dibentuk jalur pada Gambar 4.8.



Gambar 4.8 *Path* untuk UAV

Jalur pada Gambar 4.8 akan menjadi referensi untuk UAV dan juga digunakan sebagai referensi untuk menghitung *cross track error*.

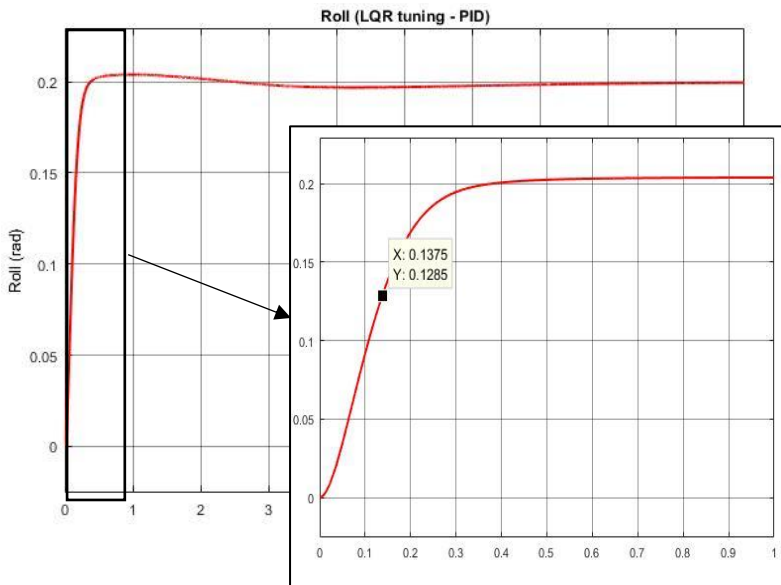
4.4 Pengujian Pengendali LQR *tuning* – PID

Pengujian pengendali LQR *tuning* – PID dilakukan dengan memberikan input referensi berupa bilangan konstan. Di dalam simulasi ini, diberikan nilai referensi untuk δ_a , δ_e , dan δ_r sebesar 0.2 rad. Pengujian ini dilakukan untuk memeriksa kemampuan kontroler dalam mencapai nilai referensi. Dan juga apakah kontroler dapat bekerja dengan baik. Dengan nilai matriks Q dan R sama seperti Persamaan (3.11), (3.12) dan (3.13) didapatkan nilai *gain* PID dari hasil *tuning* LQR seperti Tabel 4.2.

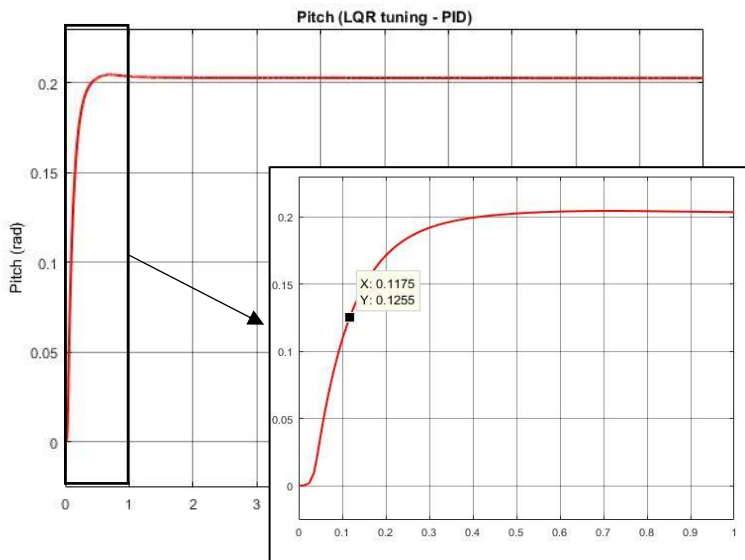
Tabel 4.2 Nilai gain PID dari tuning LQR

Defleksi	Kp	Ki	Kd
<i>Roll</i>	1470,6	447,2	171,8
<i>Pitch</i>	2832,2	31,6	240,1
<i>Yaw</i>	2240,6	31,6	313,4

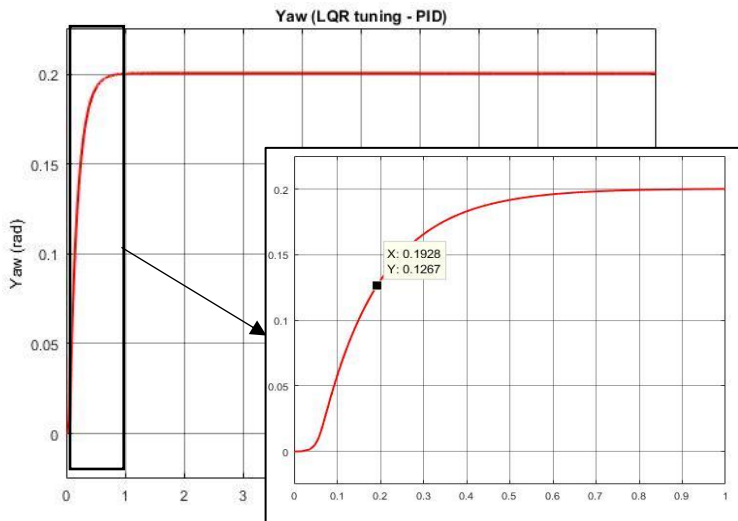
Hasil simulasi dapat dilihat pada Gambar 4.9, 4.10, dan 4.11, serta Tabel 4.3.



Gambar 4.9 Respon *roll* pengendali LQR *tuning* – PID referensi 0,2



Gambar 4.10 Respon *pitch* pengendali LQR *tuning* – PID referensi 0,2



Gambar 4.11 Respon *yaw* pengendali LQR *tuning* – PID referensi 0,2

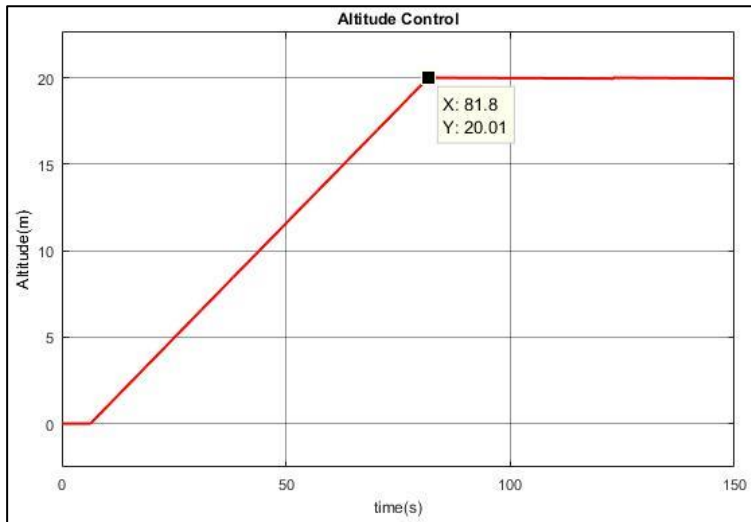
Tabel 4.3 Data hasil pengujian untuk spesifikasi kontroler

Roll (Desired $\tau = 0,1$)				
Reference	Actual τ	Error	Tolerance	Status
0,1	0,135	0,035	0,1s	Baik
0,2	0,137	0,037		Baik
0,3	0,133	0,033		Baik
0,4	0,136	0,036		Baik
0,5	0,134	0,034		Baik
Pitch (Desired $\tau = 0,1$)				
Reference	Actual τ	Error	Tolerance	Status
0,1	0,12	0,02	0,1s	Baik
0,2	0,117	0,017		Baik
0,3	0,114	0,014		Baik
0,4	0,113	0,013		Baik
0,5	0,113	0,013		Baik
Yaw (Desired $\tau = 0,1$)				
Reference	Actual τ	Error	Tolerance	Status
0,1	0,198	0,098	0,1s	Baik
0,2	0,192	0,092		Baik
0,3	0,185	0,085		Baik
0,4	0,184	0,084		Baik
0,5	0,179	0,079		Baik

Dari nilai τ yang diinginkan sebesar 0,1s, untuk *rise time*, secara ideal, pada saat waktu berjalan 0,1s, nilai grafik sudah berada pada nilai 63,2% dari nilai referensi. Namun, setelah dilakukan pengujian dengan toleransi sebesar 0,1s didapatkan hasil seperti Tabel 4.3. Jika dilihat, dengan matriks Q, R dan gain yang dipilih, serta toleransi 0,1s, semua pengendali dapat bekerja dengan baik.

4.5 Pengujian *Altitude Control*

Di dalam pengendali ketinggian, pengujian dilakukan dengan cara menambahkan outer loop di pengendalian pitch. Dikarenakan elevator pada pesawat digunakan untuk mengatur ketinggian pesawat. Hal ini dilakukan agar mendapatkan konfigurasi *cascade* dengan *outer loop* berfungsi untuk mengatur ketinggian dan *inner loop* digunakan untuk mengatur sudut pitch. Referensi ketinggian yang diberikan ialah +20m. Didapatkan respon sistem seperti pada Gambar 4.12 berikut serta data pada Tabel 4.4.



Gambar 4.12 Respon pengendali ketinggian dengan referensi +20m

Tabel 4.4 Hasil pengujian ketinggian

Referensi ketinggian	Waktu yang dibutuhkan
+5m	25,26s
+10m	44,09s
+15m	62,94s
+20m	81,8 s

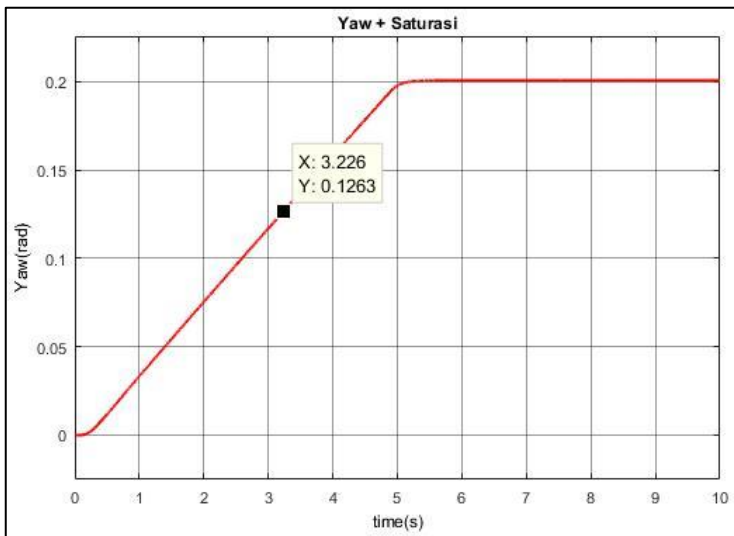
Simulasi dijalankan dengan waktu 150 detik. Jika dilihat dari respon sistem, pesawat baru dapat mencapai ketinggian yang diinginkan pada saat waktu simulasi berjalan 81,8s. Hal ini dikarenakan ditambahkan saturasi sebesar ± 0.4 rad atau $\pm 22,91$ derajat. Hal ini dilakukan karena defleksi elevator maksimal dari pesawat adalah ± 25 derajat[8]. Namun, pada simulasi ini ditambahkan saturasi agar tidak mencapai defleksi maksimum. Hal ini menyebabkan respon sistem menjadi lebih lama namun tidak akan memaksa pesawat untuk melakukan *maneuver climbing* dengan sudut yang curam. Penambahan saturasi juga dilakukan karena tipe pengendali yang digunakan ialah pengendali proporsional. Jika tidak ditambahkan saturasi, pada saat awal simulasi berjalan di mana nilai *feedback* masih bernilai 0 yang akan mengakibatkan nilai error sama

dengan nilai referensi, sinyal kontrol yang akan diberikan ke pengendali pada *inner loop* akan mencapai 2000 karena *gain* proporsional yang digunakan bernilai $K_p=100$.

4.6 Pengujian Sistem *Path Following*

Dalam sistem penelusuran jalur atau *path following*, pengujian dilakukan dengan memeriksa apakah pesawat dapat mengikuti jalur dan *waypoint* yang diberikan oleh *waypoint planner*.

Kedua penambahan saturasi pada yaw dilakukan untuk memperlambat respon yaw dalam mencapai referensi yang diinginkan. Karena, jika tidak diperlambat, maka ketika simulasi dijalankan akan terdapat singularitas dari komputasi pada perangkat lunak MATLAB yang menyebabkan terhentinya simulasi. Hal ini menyebabkan respon yaw yang cukup lama untuk mencapai titik referensinya. Hasil respon yang baru bisa dilihat pada Gambar 4.13.



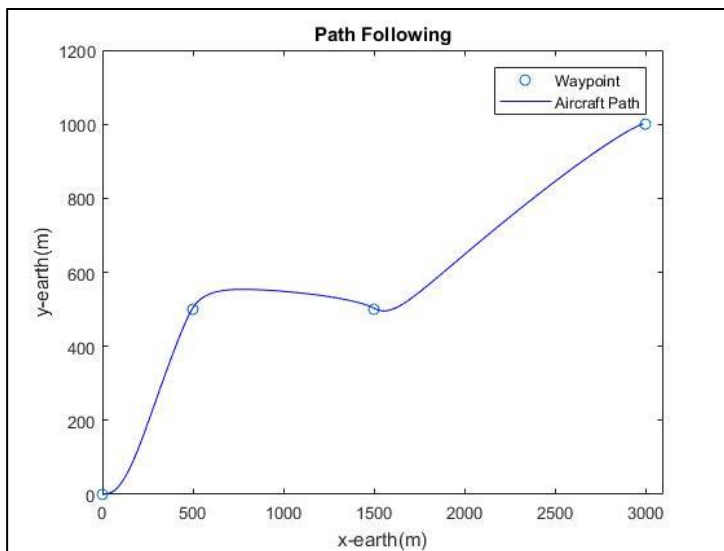
Gambar 4.13 Respon yaw setelah penambahan saturasi dengan referensi 0,2

Dapat dibandingkan perubahan *time constant* (τ) pada Tabel 4.5

Tabel 4.5 Perubahan τ pada respon yaw

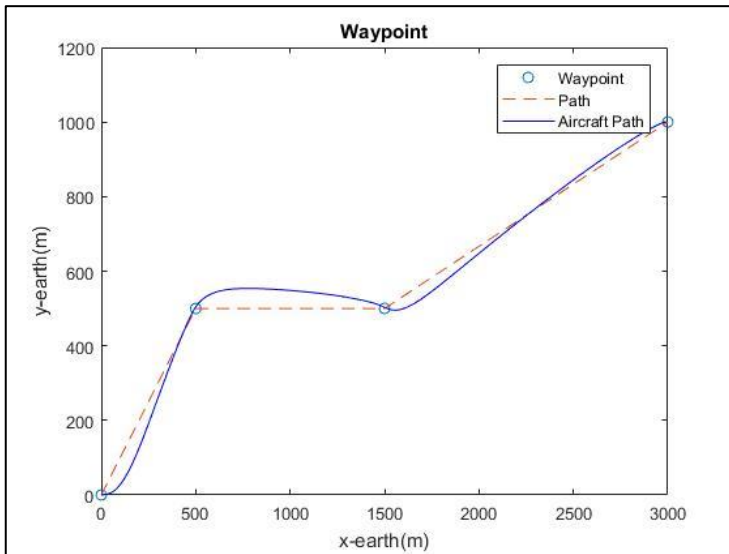
Sudut	Time constant (τ)
Yaw	0,192s
Yaw + Saturasi	3,226s
Perubahan	3,034s

Setelah komponen – komponen yang bersangkutan sudah dimodifikasi, dapat dilakukan pengujian sistem penelusuran jalur atau *path following*. Didapatkan hasil simulasi seperti ditunjukkan pada Gambar 4.8, Gambar 4.9 berikut.

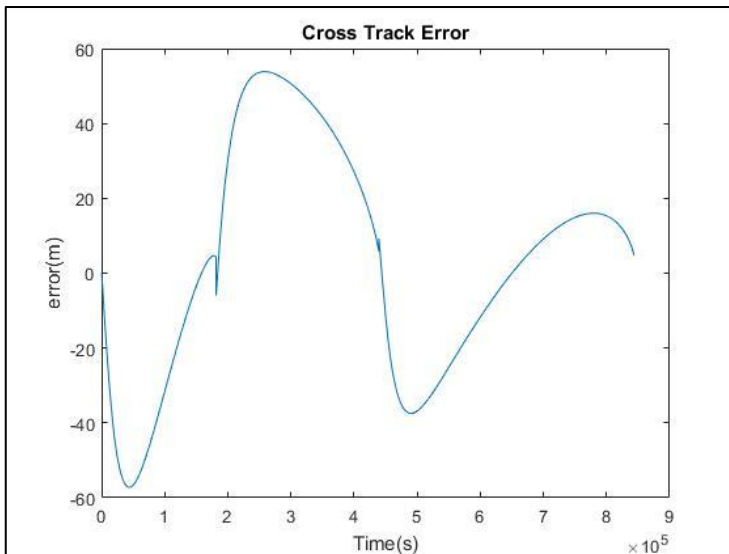


Gambar 4.14 Sistem *path following*

Jika dibandingkan dengan path yang telah dibuat oleh path generator, maka hasilnya dapat dilihat pada Gambar 4.15.



Gambar 4.15 Perbandingan jalur UAV dengan *path generator*



Gambar 4.16 Grafik *cross track error*

Pada Gambar 4.15, terlihat jelas bahwa ketika pesawat ingin bergerak dari *waypoint* pada titik (500,500) menuju titik (1500,500), pergerakannya sama sekali tidak melewati jalur ideal yang diberikan oleh *waypoint planner*. Hal ini akan menyebabkan besarnya nilai *cross track error* yang ditunjukkan pada Gambar 4.15.

Cross track error akan bernilai paling besar saat ketika meninggalkan *waypoint* untuk menuju *waypoint* berikutnya. Pada grafik *cross track error*, setiap grafik mengalami *zero crossing* saat mencapai *waypoint* tujuan. Nilai *Root-Mean-Square-Error* (RMSE) dari simulasi ini adalah 31,1735.

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

Bab ini berisi tentang kesimpulan dari proses perancangan pengendalian UAV untuk sistem penelusuran jalur beserta dengan saran untuk pengembangan kedepannya.

5.1 Kesimpulan

Dari simulasi pengendalian UAV pada sistem penelusuran jalur dapat diambil beberapa kesimpulan sebagai berikut.

1. Pengujian model dinamika dari pesawat sudah berjalan sesuai dengan seharusnya. Hal ini dapat dilihat bahwa ketika semua input dijadikan nol, pergerakan pesawat akan melakukan gerak jatuh bebas di mana tidak adanya *angle of attack* yang menyebabkan ketinggian pesawat yang semakin menurun seiring berjalannya waktu.
2. Pada pengujian pengendali LQR *tuning* – PID, respon sistem dari roll dan pitch memiliki nilai yang belum cukup sesuai dengan desain yaitu memiliki *time constant* 0,1. Respon sudut roll, pitch, dan yaw masing – masing memiliki *time constant* 0,137; 0,117; dan 0,192. Berdasarkan toleransi yang digunakan sebesar 0.1, maka semua respon sistem bisa dibilang telah bekerja dengan baik. Ketika menggunakan pengendali LQR *tuning* – PID, variabel tuning adalah nilai komponen – komponen matriks Q dan R yang akan merubah nilai gain proporsional, integral, dan derivative. Dikarenakan matriks Q dan R didapatkan dengan metode *trial and error*, hal ini memungkinkan bahwa matriks Q dan R yang penulis gunakan ialah bukan nilai yang optimum untuk sistem ini.
3. Di dalam pengujian sistem path following, pesawat sudah dapat mengikuti *waypoint* yang diberikan cukup baik. Namun, jika dibandingkan dengan jalur ideal yang diberikan oleh Gambar 4.4, masih terdapat banyak perbedaan yang terjadi. Hal ini dibuktikan dengan besarnya nilai *cross track error* yang ditunjukkan pada Gambar 4.11 dan nilai *Root-Mean-Square*-

Error (RMSE) yang dihasilkan dari grafik *cross track error* sebesar 31, 1735. Hal ini bisa disebabkan oleh respon yaw yang harus diberikan saturasi untuk memperlambat respon agar tidak terjadi singularitas saat menjalankan komputasi di simulasi pada perangkat lunak MATLAB 2017a yang penulis gunakan. Untuk keseluruhannya, dapat disimpulkan bahwa penggunaan pengendali *LQR tuning* – PID dapat digunakan sebagai kontroler dalam sistem *path following*. Namun, untuk meningkatkan kualitas dari pergerakannya, masih terdapat banyak hal yang harus ditingkatkan.

5.2 Saran

Untuk pengembangan kedepan, terdapat beberapa saran yang diberikan, antara lain:

1. Pada saat ingin mendapatkan nilai matriks Q dan R, dapat dilakukan dengan metode lainnya contohnya algoritma genetika agar mendapatkan nilai yang optimal.
2. Menggunakan *guidance* yang lebih cocok untuk metode ini agar UAV dapat mengikuti jalur lebih baik.

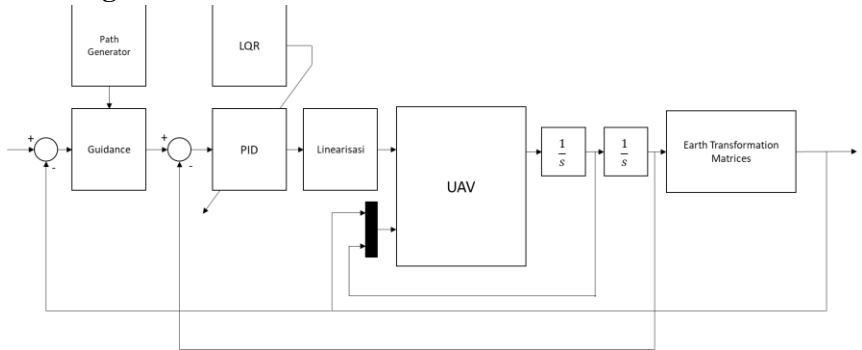
DAFTAR PUSTAKA

- [1] R. W. Beard dan T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*, New Jersey: Princeton University, 2012.
- [2] L. M. Argentim, W. C. Rezende dan P. E. Santos, "PID, LQR, and LQR-PID on Quadcopter Platform," dalam *IEEE*, São Bernardo do Campo, 2013.
- [3] S. Mukhopadhyay, "PID equivalent of optimal regulator," *Electronics Letter*, vol. 14, no. 25, pp. 821-822, 1978.
- [4] R. E. A.K., M. Rameli, E. Iskandar dan M. Baihaqi, "Linearization and decoupling controller for quadruple tank," dalam *ISITIA*, Surabaya, 2017.
- [5] I. H. Johanssen, *Autopilot Design for Unmanned Aerial Vehicles*, Trondheim: Norwegian University of Science and Technology, 2012.
- [6] M. R. Prasetyo, *Perancangan Kendali Optimal LQT untuk Pengendalian dan Pemanduan pada Rudal*, Surabaya: Teknik Elektro Institut Teknologi Sepuluh Nopember, 2018.
- [7] Mathworks, "Designing Cascade Control System with PI Controllers," MATLAB & Simulink, 2018. [Online]. Available: <https://www.mathworks.com/help/control/ug/designing-cascade-control-system-with-pi-controllers.html>. [Diakses 16 Juli 2018].
- [8] M. Sadraey, *Aircraft Design: A Systems Engineering Approach*, New Hampshire: Daniel Webster College, 2013.
- [9] NASA, "Aircraft Rotation: Body Axis," 2015. [Online]. Available: <https://www.grc.nasa.gov/WWW/k-12/airplane/rotations.html>. [Diakses 16 Juli 2018].
- [1] U. o. Michigan, "Introduction: PID Controller Design," 1985.
- [0] [Online]. Available: <http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=ControlPID>. [Diakses 16 Juli 2018].

Halaman ini sengaja dikosongkan

LAMPIRAN A

A1. Diagram Blok Sistem



A2. Program Dinamika UAV

```
function [udot,vdot,wdot,pdot,qdot,rdot,medot]
= dynamicskinematics(Def,TR,Euler)
%info
%{
Output:
udot = x-axis acceleration;   pdot = roll-axis
angular acceleration;   medot = transformation
matrix to earth frame;
vdot = y-axis acceleration;   qdot = pitch-axis
angular acceleration;
wdot = z-axis acceleration;   rdot = yaw-axis
angular acceleration;
Input:
Def = roll, pitch, yaw, throttle control
signal;
BFrame = movement on body frame;
Euler = roll, pitch, yaw feedback;
Wind = wind input;
%}
da=Def(1);
de=Def(2);
dr=Def(3);
```

```

u=TR(1);
v=TR(2);
w=TR(3);
p=TR(4);
q=TR(5);
r=TR(6);
roll=Euler(1);
pitch=Euler(2);
yaw=Euler(3);
%compute air data
ur=u; vr=v; wr=w;
Va=sqrt(ur^2+vr^2+wr^2);
alpha=atan2(wr,ur);
B=asin(vr/Va);
%parameter & coeficient
M=1.56;
%M=5;
g=9.81;
Ix=0.1147;
Iy=0.0576;
Iz=0.1712;
Ixz=0.0015;
S=0.2589;
b=1.4224;
c=0.3302;
Sprop=0.0314;
%Sprop=0.5;
Cprop=1;
Kmotor=20;
R=1.2682;
dt=0;
KTp=0;
KOmega=0;
%translational components
CLo=0.09167;
CLa=3.5026;
CLA=CLo+CLa*alpha;
CLq=2.8932;
CLde=0.2724;
CDo=0.01613;

```

```

CDA=0.2108;
CDA=CDo+CDA*alpha;
CDq=0;
CDde=0.3045;
Cyo=0;
Cyb=-0.07359;
Cyp=0;
Cyr=0;
Cyda=0;
Cydr=-0.17;
%x-axis
Cxa=-CDA*cos(alpha)+CLA*sin(alpha);
Cxqa=-CDq*cos(alpha)+CLq*sin(alpha);
Cxdea=-CDde*cos(alpha)+CLde*sin(alpha);
%z-axis
Cza=-CDA*sin(alpha)-CLA*cos(alpha);
Czqa=-CDq*sin(alpha)-CLq*cos(alpha);
Czdea=-CDde*sin(alpha)-CLde*cos(alpha);
%rotational components
Cmo=-0.02338;
Cma=-0.5675;
Cmq=-1.3990;
Cmde=-0.3254;
Clo=0;
Clb=-0.02854;
Clp=-0.3209;
Clr=0.03066;
Clda=0.1682;
Clldr=0.105;
Cno=0;
Cnb=-0.0040;
Cnp=-0.01297;
Cnr=-0.00434;
Cnda=-0.00328;
Cndr=-0.032;
%Inertial Product
A=Ix*Iz-Ixz^2;
A1=(Ixz*(Ix-Iy+Iz))/A;
A2=(Iz*(Iz-Iy)+Ixz^2)/A;
A3=Iz/A;

```

```

A4=Ixz/A;
A5=(Iz-Ix)/A;
A6=Ixz/A;
A7=((Ix-Iy)*Ix+Ixz^2)/A;
A8=Ix/A;
%roll
Cpo=A3*Clo+A4*Cno;
Cpb=A3*Clb+A4*Cnb;
CpP=A3*Clp+A4*Cnp;
Cpr=A3*Clr+A4*Cnr;
Cpda=A3*Cllda+A4*Cnda;
Cpdr=A3*Clldr+A4*Cndr;
%yaw
Cro=A4*Clo+A8*Cno;
Crb=A4*Clb+A8*Cnb;
Crp=A4*Clp+A8*Cnp;
Crr=A4*Clr+A8*Cnr;
Crda=A4*Cllda+A8*Cnda;
Crdr=A4*Clldr+A8*Cndr;

%mathematical process
%translational
udot=r*v-q*w-
g*sin(pitch)+(R/2*M)*Va^2*S*(Cxa+Cxqa*((c*q)/(
2*Va))+Cxdea*de)+(R/2*M)*Sprop*Cprop*((Kmotor*
dt)^2-Va^2);
vdot=p*w-
r*u+g*cos(pitch)*sin(roll)+(R/2*M)*Va^2*S*(Cyo
+Cyb*B+Cyp*(b*p/2*Va)+Cyr*(b*r/2*Va)+Cyda*da+C
ydr*dr);
wdot=q*u-
p*v+g*cos(pitch)*cos(roll)+(R/2*M)*Va^2*S*(Cza
+Czqa*((c*q)/(2*Va))+Czdea*de);
f1=[udot;vdot;wdot];
%rotational
pdot=(A1*p*q)-
(A2*q*r)+(R/2)*(Va^2*S*b)*(Cpo+Cpb*B+CpP*(b*p/
2*Va)+Cpr*(b*r/2*Va)+Cpda*da+Cpdr*dr);

```

```

qdot=(A5*p*r)-(A6*(p^2-
r^2))+(R/(2*Iy))*(Va^2*S*c)*(Cmo+Cma*alpha+Cmq
*(c*p/2*Va)+Cmde*de);
rdot=(A7*p*q)-
(A1*q*r)+(R/2)*(Va^2*S*b)*(Cro+Crb*B+Crp*(b*p/
2*Va)+Crr*(b*r/2*Va)+Crda*da+Crdr*dr);
f2=[pdot;qdot;rdot];
%rotation matrix
C=[cos(pitch)*cos(yaw)
sin(roll)*sin(pitch)*cos(yaw)-
cos(roll)*sin(yaw)
cos(roll)*sin(pitch)*sin(yaw)+sin(roll)*sin(ya
w)];
    cos(pitch)*sin(yaw)
sin(roll)*sin(pitch)*sin(yaw)+cos(roll)*cos(ya
w) cos(roll)*sin(pitch)*sin(yaw)-
sin(roll)*cos(yaw);
    sin(pitch) -sin(roll)*cos(pitch) -
cos(roll)*cos(pitch)];

%earth transformation matrices
%tranlation
C=[cos(pitch)*cos(yaw)
sin(roll)*sin(pitch)*cos(yaw)-
cos(roll)*sin(yaw)
cos(roll)*sin(pitch)*sin(yaw)+sin(roll)*sin(ya
w)];
    cos(pitch)*sin(yaw)
sin(roll)*sin(pitch)*sin(yaw)+cos(roll)*cos(ya
w) cos(roll)*sin(pitch)*sin(yaw)-
sin(roll)*cos(yaw);
    sin(pitch) -sin(roll)*cos(pitch) -
cos(roll)*cos(pitch)];
%rotation
D=[1 sin(roll)*tan(pitch)
cos(roll)*tan(pitch);
    0 cos(roll) -sin(roll);
    0 sin(roll)/cos(pitch)
cos(roll)/cos(pitch)];

```

```

%earth transformation
medottr=C*[u v w]';
medotrt=D*[p q r]';
medot=[medottr;medotrt]';

```

A3. Program Non-Linear State Feedback Decoupler

```

function Def =
Linearization(csd,longlatfb,rot)

%input
da=csd(1);
de=csd(2);
dr=csd(3);
u=longlatfb(1);
v=longlatfb(2);
w=longlatfb(3);
p=longlatfb(4);
q=longlatfb(5);
r=longlatfb(6);
roll=rot(1);
pitch=rot(2);
yaw=rot(3);
%compute air data
ur=u; vr=v; wr=w;
Va=sqrt(ur^2+vr^2+wr^2);
alpha=atan2(wr,ur);
B=asin(vr/Va);
%parameter & coeficient
M=1.56;
g=9.81;
Ix=0.1147;
Iy=0.0576;
Iz=0.1712;
Ixz=0.0015;
S=0.2589;
b=1.4224;
c=0.3302;
Sprop=0.0314;

```



```

Cprop=1;
Kmotor=20;
R=1.2682;
dt=1;
KTP=0;
KOmega=0;
lamda=10;
%translational components
CLO=0.09167;
CLa=3.5026;
CLA=CLO+CLa*alpha;
CLq=2.8932;
CLde=0.2724;
CDo=0.01613;
CDa=0.2108;
CDA=CDo+CDa*alpha;
CDq=0;
CDde=0.3045;
Cyo=0;
Cyb=-0.07359;
Cyp=0;
Cyr=0;
Cyda=0;
Cydr=-0.17;
%x-axis
Cxa=-CDA*cos(alpha)+CLA*sin(alpha);
Cxqa=-CDq*cos(alpha)+CLq*sin(alpha);
Cxdea=-CDde*cos(alpha)+CLde*sin(alpha);
%z-axis
Cza=-CDA*sin(alpha)-CLA*cos(alpha);
Czqa=-CDq*sin(alpha)-CLq*cos(alpha);
Czdea=-CDde*sin(alpha)-CLde*cos(alpha);
%rotational components
Cmo=-0.02338;
Cma=-0.5675;
Cmq=-1.3990;
Cmde=-0.3254;
Clo=0;
Clb=-0.02854;
Clp=-0.3209;

```

```

Clr=0.03066;
Clda=0.1682;
Cldr=0.105;
Cno=0;
Cnb=-0.0040;
Cnp=-0.01297;
Cnr=-0.00434;
Cnda=-0.00328;
Cndr=-0.032;
%Inertial Product
A=Ix*Iz-Ixz^2;
A1=(Ixz*(Ix-Iy+Iz))/A;
A2=(Iz*(Iz-Iy)+Ixz^2)/A;
A3=Iz/A;
A4=Ixz/A;
A5=(Iz-Ix)/A;
A6=Ixz/A;
A7=((Ix-Iy)*Ix+Ixz^2)/A;
A8=Ix/A;
%roll
Cpo=A3*Clo+A4*Cno;
Cpb=A3*Clb+A4*Cnb;
Cppl=A3*Clp+A4*Cnp;
Cpr=A3*Clr+A4*Cnr;
Cpda=A3*Clda+A4*Cnda;
Cpdr=A3*Cldr+A4*Cndr;
%yaw
Cro=A4*Clo+A8*Cno;
Crb=A4*Clb+A8*Cnb;
Crp=A4*Clp+A8*Cnp;
Crr=A4*Clr+A8*Cnr;
Crda=A4*Clda+A8*Cnda;
Crdr=A4*Cldr+A8*Cndr;

%process
CC=[Cpda 0 Cpdr;0 Cmde 0;Crda 0 Crdr];
CP=[-((2/R)*(1/Va^2*S*b))*((A1*p*q)-(A2*q*r)+(R/2)*(Va^2*S*b))*(Cpo+Cpb*B+Cppl*(b*p/2*Va)+Cpr*(b*r/2*Va))+lamda*p-da)];-

```

```

( (2/R) * (1/Va^2*S*b*Cmde) * (A5*p*r) - (A6* (p^2-
r^2)) + (R/ (2*Iy)) * (Va^2*S*c) * (Cmo+Cma*alpha+Cmq
* (c*p/2*Va)) + lamda*q-de) ; -
( (2/R) * (1/Va^2*S*b) * (A7*p*q) -
(A1*q*r) + (R/2) * (Va^2*S*b) * (Cro+Crb*B+Crp* (b*p/
2*Va)+Crr* (b*r/2*Va)) + lamda*r-dr] );
CD=inv(CC)*CP;
%output
dal=CD(1,:);%+lamda*p-da;
del=CD(2,:);%+lamda*q-de;
drl=CD(3,:);%+lamda*r-dr;
Def=[dal;del;drl];

```

A4. Program LQR tuning – PID

A4.1 Roll

```

function K = RollGain(R)

%parameter
lamda=10;

%matrices
Q=[20000 0 0;0 300 0;0 0 2000];
R=0.01;

%new augmented matrices
Ab=[0 1 0;0 -lamda 0;1 0 0];
Bb=[0;1;0];

%riccati equation
[X,L,G]=care(Ab,Bb,Q,R);

KR=inv(R)*Bb.'*X;

K=KR';

```

A4.2 Pitch

```
function K = PitchGain(R)

%parameter
lamda=100;

% matrices
Q=[80000 0 0;0 1000 0;0 0 10];
R=0.01;

%new augmented matrices
Ab=[0 1 0;0 -lamda 0;1 0 0];
Bb=[0;1;0];

%riccati equation
[X,L,G]=care(Ab,Bb,Q,R);

KP=inv(R)*Bb.'*X;

K=KP';
```

A4.3 Yaw

```
function K = YawGain(R)

%parameter
lamda=10;

%matrices
Q=[50000 0 0;0 1000 0;0 0 10];
R=0.01;

%new augmented matrices
Ab=[0 1 0;0 -lamda 0;1 0 0];
```

```

Bb=[0;1;0];

%riccati equation
[X,L,G]=care (Ab,Bb,Q,R);

KY=inv(R)*Bb.'*X;

K=KY';

```

A5. Program Waypoint Planner

A5.1 Path Generator

```

% Inisialisasi Guidance
Data_Waypoint = [0 0;500 500;1500 500;3000
1000];
figure;
plot(Data_Waypoint(:,1),Data_Waypoint(:,2),'o'
);
hold on;
plot(Data_Waypoint(:,1),Data_Waypoint(:,2),'--
');
title('Waypoint');
xlabel('x-earth');
ylabel('y-earth');
axis([0 3100 0 1200]);

```

A5.2 Guidance

```

function Yaw_Reference =
Guidance(x,y,Data_Waypoint)
global index
%Ambil aim point
Aim_Point = Data_Waypoint(index,:);
% Menghitung Delta
dx = Aim_Point(1)-x;
dy = Aim_Point(2)-y;
Length_Lxy = sqrt(dx^2+dy^2);

```

```

%Merubah aim point saat toleransi terpenuhi
Tolerance = 12;
if Length_Lxy<Tolerance
    index = index+1;
end
%Menghitung yaw reference
Yaw_Reference = atan2(dy,dx);

```

A6. Program Cross Track Error

```

clc
load('waypoint.mat')
load('XY.mat')
%Waypoint tolerance
tolerance = 12;
%index Waypoint
i = 1
%index Data
Data = 1;

for Data=1:length(x/10)
    absx = abs(x(Data,1)-Data_Waypoint(i,1));
    absy = abs(y(Data,1)-Data_Waypoint(i,2));
    if absx<tolerance && absy<tolerance
        i=i+1;
    end
    etatoPath(Data,:) = [x(Data,1) y(Data,1)
i-1 i];
end

% Cross track error
Data= 1
for Data=1:length(x)-10
    dx = x(Data,1)-
Data_Waypoint(etatoPath(Data,3),1);
    dy = y(Data,1)-
Data_Waypoint(etatoPath(Data,3),2);
    Theta = atan2(dy,dx);

```

```

        dx_data =
Data_Waypoint(etatoPath(Data,4),1)-
Data_Waypoint(etatoPath(Data,3),1);
        dy_data =
Data_Waypoint(etatoPath(Data,4),2)-
Data_Waypoint(etatoPath(Data,3),2);
        Theta_data = atan2(dy_data,dx_data);
        Length_data = sqrt(dx^2+dy^2);
        Theta_s = Theta_data-Theta;
        d = Length_data*sin(-Theta_s);
        etatoPathCTE(Data,:) = [x(Data,1)
y(Data,1) i-1 i d ];
end

plot(tout,etatoPathCTE(:,5))

```

Halaman ini sengaja dikosongkan

RIWAYAT HIDUP PENULIS



Muhamad Dhyaksa Khairana lahir di Jakarta pada tanggal 2 Mei 1996. Penulis merupakan putra pertama dari dua bersaudara. Putra dari pasangan Anden Arief Wicaksana dan Noordini Rianti ini memulai jenjang pendidikannya di TK Patra 4 Jakarta, lalu dilanjutkan ke SDN Gunung 05 Mexico Pagi Jakarta. Lalu penulis melanjutkan jenjang pendidikannya ke SMP 11 Jakarta dan berlanjut ke SMA 82 Jakarta hingga lulus pada tahun 2014. Selanjutnya, penulis melanjutkan pendidikannya ke jenjang perguruan tinggi di Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember. Selama masa perkuliahan, penulis aktif sebagai asistem dari Laboratorium Sistem dan Sibernetika. Selama menjadi asisten, penulis aktif sebagai Koordinator Asisten untuk asistem di Laboratorium Sistem dan Sibernetika yang mewakili ruangan AJ204 dan B405. Penulis dapat dihubungi melalui surel mdhyaksa.khairana@gmail.com.

Halaman ini sengaja dikosongkan